

Package ‘SGP’

April 12, 2013

Type Package

Title An R Package for the Calculation and Visualization of Student Growth Percentiles & Percentile Growth Trajectories.

Version 1.1-0.0

Date 2013-4-12

Author Damian W. Betebenner <dbetebenner@nciea.org>, Adam Van Iwaarden <vaniwaarden@colorado.edu>, Ben Domingue <ben.domingue@gmail.com>, and Yi Shang <shangyi@gmail.com> with contributions from Dr. Jonathan Weeks, Dr. Yi Shang, John Stewart, Dr. Jinnie Choi, Dr. Xin Wei, and Hi Shin Shim. With special thanks to State Department of Education analysts: Dr. Xiaoyuan Tan (Arizona), Dr. Carrie Giovannini (Arizona), Dr. Sarah Polasky (Arizona), Rebecca Gau (Arizona), Dr. Bill Bonk (Colorado), Marie Huchton (Colorado), Dr. Allison Timberlake (Georgia), Dr. Melissa Fincher (Georgia), Kiran Athota (Georgia), Travis Allen (Georgia), Glenn Hirata (Hawaii), Glenn Nochi (Hawaii), Joshua Lee (Hawaii), Ayaka Nukui (Idaho), Dr. Carissa Miller (Idaho), Dr. Wes Bruce (Indiana), Robert Hochsegang (Indiana), Tony Moss (Kansas), Dr. Xuewen Sheng (Kansas), Dr. Kathy Flanagan (Massachusetts), Robert Lee (Massachusetts), Joe DeCastra (Mississippi), Ken Thompson (Mississippi), Soo Yeon Cho (Missouri), Dr. Jeff Halsell (Nevada), Dr. Selcuk Ozdemir (Nevada), Roger Silva (Nevada), Deb Wiswell (New Hampshire), Katya Levitan-Reiner (New Haven Public Schools), Catherine McCaslin (New Haven Public Schools), Joshua Marland (New York), W Joshua Rew (Oregon), Jason Becker (Rhode Island), Dr. Jessica Brown (Rhode Island), Ana Karantonis (Rhode Island), Dr. Deborah Jonas (Virginia), Juan D’Brot (West Virginia), Dr. Deb Came (Washington), Krissy Johnson (Washington), Nate Hixson (West Virginia), Daniel Bush (Wisconsin), Justin Meyer (Wisconsin), Joseph Newton (Wisconsin), Dr. Philip Olsen (Wisconsin), Nick Stroud (Wisconsin), Brendan Hough (University of Melbourne, Australia).

Maintainer Damian W. Betebenner <dbetebenner@nciea.org>

Depends R (>= 2.15), SGPdata (>= 7.0-0), methods

Suggests randomNames (>= 0.0-5), parallel

Imports Cairo, colorspace, data.table (>= 1.8.6), doParallel, foreach, grid, gridBase, iterators, plyr, quantreg, reshape2, RSQLite, sn, SparseM, splines

Description Functions to calculate student growth percentiles and percentile growth projections/trajectories for students using large scale, longitudinal assessment data. Functions use quantile regression to estimate the conditional density associated with each student's achievement history. Percentile growth projections/trajectories are calculated using the coefficient matrices derived from the quantile regression analyses and specify what percentile growth is required for students to reach future achievement targets.

SystemRequirements (PDF)LaTeX (<http://www.latex-project.org/>) with 'pdfpages' package for studentGrowthPlot option in visualizeSGP to bind together student growth plots into school catalogs.

URL <https://github.com/SchoolView/SGP>, <http://schoolview.github.com/SGP/>, <http://cran.r-project.org/web/packages/SGP/>

LazyData Yes

LazyDataCompression xz

License CC BY-SA 3.0 US + file LICENSE

ByteCompile TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2013-04-12 16:48:16

R topics documented:

SGP-package	3
abcSGP	4
analyzeSGP	8
baselineSGP	15
bubblePlot	17
bubblePlot_Styles	22
capwords	24
combineSGP	25
createKnotsBoundaries	27
gofSGP	28

growthAchievementPlot	29
outputSGP	31
prepareSGP	33
SGP-class	34
SGPstateData	36
splineMatrix-class	36
studentGrowthPercentiles	38
studentGrowthPlot	47
studentGrowthPlot_Styles	49
studentGrowthProjections	51
summarizeSGP	56
testSGP	60
updateSGP	61
visualizeSGP	63
Index	70

SGP-package	<i>SGP: An R Package for the Calculation and Visualization of Student Growth Percentiles & Percentile Growth Trajectories.</i>
-------------	--

Description

SGP contains classes and functions to calculate student growth percentiles and percentile growth projections/trajectories following methodology found in Betebenner (2008, 2009). The package contains two primary functions, [studentGrowthPercentiles](#) and [studentGrowthProjections](#), and numerous higher level functions that make use of them including: [prepareSGP](#), [analyzeSGP](#), [combineSGP](#), [summarizeSGP](#), [visualizeSGP](#) and [outputSGP](#). These functions are used to calculate and visualize student growth percentiles and percentile growth projections/trajectories for students using large scale, longitudinal assessment data. These norm- and criterion-referenced growth values are currently used in a number of states for many purposes including diagnostic and accountability. The functions employ quantile regression (using the [quantreg](#) package) to estimate the conditional density for current achievement using each student’s achievement history. Percentile growth projections/trajectories are calculated using the coefficient matrices derived from the student growth percentile analyses. These quantities are summarized in a variety of ways to describe student growth. As of the 0.2-0.0 release, the package also includes the graphics functions [bubblePlot](#), [studentGrowthPlot](#), and [growthAchievementPlot](#) to produce high quality graphical representations associated with the student growth percentile analyses.

Details

Package:	SGP
Type:	Package
Version:	1.1-0.0
Date:	2013-4-12
License:	CC BY-SA 3.0 US + file LICENSE
LazyLoad:	yes

Calculation of student growth percentiles and percentile growth trajectories/projections is typically performed by grade and subject. Data for growth percentile calculation must be specifically formatted. See [sgpData](#) for an example data set. Batch R syntax for performing analyses across all grades and years is provided in the examples of the [studentGrowthPercentiles](#) and [studentGrowthProjections](#) using the higher level functions [prepareSGP](#), [analyzeSGP](#), [combineSGP](#), [summarizeSGP](#), and [visualizeSGP](#).

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>, Adam Van Iwaarden <Vaniwaarden@colorado.edu>, Ben Domingue <ben.domingue@gmail.com> and Yi Shang <shangyi@gmail.com>

References

- Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.
- Betebenner, D. W. (2009). Norm- and criterion-referenced student growth. *Educational Measurement: Issues and Practice*, 28(4):42-51.
- Betebenner, D. W. (2008). Toward a normative understanding of student growth. In K. E. Ryan & L. A. Shepard (Eds.), *The Future of Test Based Accountability* (pp. 155-170). New York: Routledge.
- Koenker, R. (2005). *Quantile regression*. Cambridge: Cambridge University Press.

abcSGP

Perform 5 step sequence: [prepareSGP](#), [analyzeSGP](#), [combineSGP](#), [summarizeSGP](#), and [visualizeSGP](#)

Description

Utility function to perform sequence of 6 steps going from data preparation, [prepareSGP](#), SGP data analysis, [analyzeSGP](#), data combining, [combineSGP](#), data summarization, [summarizeSGP](#), data visualization [visualizeSGP](#) and data output [outputSGP](#).

Usage

```
abcSGP(sgp_object,
state=NULL,
steps=c("prepareSGP", "analyzeSGP", "combineSGP", "summarizeSGP", "visualizeSGP", "outputSGP"),
years=NULL,
content_areas=NULL,
grades=NULL,
prepareSGP.var.names=NULL,
sgp.percentiles=TRUE,
sgp.projections=TRUE,
sgp.projections.lagged=TRUE,
sgp.percentiles.baseline=TRUE,
```

```

sgp.projections.baseline=TRUE,
sgp.projections.lagged.baseline=TRUE,
sgp.use.my.coefficient.matrices=NULL,
simulate.sgps=TRUE,
calculate.simex=NULL,
parallel.config=NULL,
save.intermediate.results=FALSE,
save.old.summaries=FALSE,
sgPlot.demo.report=FALSE,
sgp.summaries=NULL,
summary.groups=NULL,
data_supplementary=NULL,
confidence.interval.groups=NULL,
plot.types=c("bubblePlot", "studentGrowthPlot", "growthAchievementPlot"),
verbose.output=FALSE)

```

Arguments

<code>sgp_object</code>	A list containing LONG formatted data. See sgpData_LONG for an exemplar. NOTE: Data preparation must be meticulous to utilize this enhanced functionality.
<code>state</code>	Acronym indicating state associated with the data for access to embedded knot and boundaries, cutscores, CSEMs, and other relevant state level data.
<code>steps</code>	Vector containing all or some subset of prepareSGP , analyzeSGP , combineSGP , summarizeSGP , visualizeSGP indicating what steps the user wants accomplished. Default is to perform all steps.
<code>years</code>	A vector indicating year(s) in which to produce student growth percentiles and/or student growth projections/trajectories. If missing the function will use the data to infer the year(s) in which to perform growth percentile analyses based upon the assumption of having at least three years of panel data for analyses.
<code>content_areas</code>	A vector indicating content area(s) in which to produce student growth percentiles and/or student growth projections/trajectories. If missing the function will use the data to infer the content area(s) available for analyses.
<code>grades</code>	A vector indicating grades for which to calculate student growth percentiles and/or student growth projections/trajectories. If missing the function will use the data to infer all the grade progressions for student growth percentile and student growth projections/trajectories analyses.
<code>prepareSGP.var.names</code>	list supplied to <code>prepareSGP</code> mapping provided variable names to variable names required as part of the SGP package. See prepareSGP for more details. Defaults to NULL.
<code>sgp.percentiles</code>	Boolean variable indicating whether to calculate student growth percentiles. Defaults to TRUE.
<code>sgp.projections</code>	Boolean variable indicating whether to calculate student growth projections. Defaults to TRUE.

<code>sgp.projections.lagged</code>	Boolean variable indicating whether to calculate lagged student growth projections often used for growth to standard analyses. Defaults to TRUE.
<code>sgp.percentiles.baseline</code>	Boolean variable indicating whether to calculate baseline student growth percentiles and/or coefficient matrices. Defaults to TRUE.
<code>sgp.projections.baseline</code>	Boolean variable indicating whether to calculate baseline student growth projections. Defaults to TRUE.
<code>sgp.projections.lagged.baseline</code>	Boolean variable indicating whether to calculate lagged baseline student growth projections. Defaults to TRUE.
<code>sgp.use.my.coefficient.matrices</code>	Boolean variable indicating whether to use embedded coefficient matrices to calculate SGPs. One should be careful to remove previously calculated SGPs prior to recalculating SGPs.
<code>simulate.sgps</code>	Boolean variable indicating whether to simulate SGP values for students based on test-specific Conditional Standard Errors of Measurement (CSEM). Test CSEM data must be available for simulation. Must be set to TRUE for confidence interval construction. Defaults to TRUE in abcSGP only.
<code>calculate.simex</code>	A character state acronymn or list including state/csem variable, simulation.iterations, lambda and extrapolation method. Default to NULL, no simex calculations done.
<code>parallel.config</code>	A named list with, at a minimum, two elements indicating 1) the BACKEND package to be used for parallel computation and 2) the WORKERS list to specify the number of processors to be used in each major analysis. The BACKEND element can be set = to FOREACH or PARALLEL. Please consult the manuals and vignettes for information of these packages! The analyzeSGP help page contains more thorough explanation and examples of the <code>parallel.config</code> setup. The <code>parallel.config</code> list is passed to analyzeSGP , summarizeSGP and visualizeSGP . The WORKERS list can accordingly contain elements for PERCENTILES, PROJECTIONS, LAGGED_PROJECTIONS, BASELINE_MATRICES, BASELINE_PERCENTILES for analyzeSGP , SUMMARY for summarizeSGP and GA_PLOTS and SG_PLOTS for visualizeSGP . See those functions help pages for details.
<code>save.intermediate.results</code>	Should intermediate results of abcSGP be saved after each of prepareSGP , analyzeSGP , combineSGP , and summarizeSGP . Default is FALSE.
<code>save.old.summaries</code>	A boolean argument (defaults to FALSE which will delete the @Summary slot before creating new summaries) indicating whether the call to summarizeSGP should save existing summaries in the @Summary slot.
<code>sgPlot.demo.report</code>	A boolean variable (defaults to FALSE) indicating whether to produce only the demonstration student report catalog. Default is to produce reports for entire current year data.

- `sgp.summaries` A list giving the summaries requested for each group analyzed based upon the `summary.group` argument. Default is NULL allowing the [summarizeSGP](#) function to produce the list of summaries automatically.
- `summary.groups` A list consisting of 8 types of groups across which all summaries are taken: `institution`, `content`, `time`, `institution_type`, `institution_level`, `demographic`, and `institution_inclusion`. Summaries generated in `summarizeSGP` are for all possible combinations of the 8 types of group. See documentation for [summarizeSGP](#) for more detail.
- `data_supplementary`
A list argument (or NULL, the default) providing additional multiple membership lookup tables for `summarizeSGP`. See `sgpData_INSTRUCTOR_NUMBER` for an example. Supplied data is embedded in the `@Data_Supplementary` slot.
- `confidence.interval.groups`
A subset of the groups provided in the `summary.groups` argument indicating which groups to provide confidence intervals for. See documentation for [summarizeSGP](#) for more detail.
- `plot.types` A character vector passed to [visualizeSGP](#) indicating the types of plots to produce. Currently supported plots include [bubblePlots](#), [studentGrowthPlots](#), and [growthAchievementPlots](#).
- `verbose.output` A boolean argument indicating whether the function should output verbose diagnostic messages.

Value

Function returns a list containing the input long data set in the `Student` slot as a `data.table` keyed using `VALID_CASE`, `CONTENT_AREA`, `YEAR`, `ID`, SGP results including student growth percentile and student growth projections/trajectories in the `SGP` slot, and summary results in the `Summary` slot.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <vaniwaarden@colorado.edu>

See Also

[prepareSGP](#), [analyzeSGP](#), [combineSGP](#), [summarizeSGP](#), [studentGrowthPercentiles](#), and [studentGrowthProjections](#)

Examples

```
## Not run:

## Runs all 5 steps
Demonstration_SGP <- abcSGP(sgp_object=sgpData_LONG, state="DEMO")

## Or letting the function detect the state

Demonstration_SGP <- sgpData_LONG
Demonstration_SGP <- abcSGP(Demonstration_SGP)
```

```

###
### Example uses of the parallel.config argument
###

Demonstration_SGP <- sgpData_LONG
Demonstration_SGP <- abcSGP(Demonstration_SGP,
parallel.config=list(
  BACKEND="PARALLEL", TYPE="SOCK",
  WORKERS=list(
    PERCENTILES=8, BASELINE_PERCENTILES=8, PROJECTIONS=7, LAGGED_PROJECTIONS=6,
    SUMMARY=8,
    GA_PLOTS=8, SG_PLOTS=8)
)
)

## End(Not run)

```

analyzeSGP	<i>Analyze student data to produce student growth percentiles and student growth projections</i>
------------	--

Description

Utility function/exemplar used to produce student growth percentiles and student growth projections using long formatted data like that provided by [prepareSGP](#).

Usage

```

analyzeSGP(sgp_object,
  state=NULL,
  years=NULL,
  content_areas=NULL,
  grades=NULL,
  sgp.percentiles=TRUE,
  sgp.projections=TRUE,
  sgp.projections.lagged=TRUE,
  sgp.percentiles.baseline=TRUE,
  sgp.projections.baseline=TRUE,
  sgp.projections.lagged.baseline=TRUE,
  sgp.percentiles.baseline.max.order=3,
  sgp.projections.baseline.max.order=3,
  sgp.projections.lagged.baseline.max.order=3,
  sgp.minimum.default.panel.years=3,
  sgp.use.my.coefficient.matrices=NULL,
  simulate.sgps=TRUE,

```



```

calculate.simex=NULL,
      goodness.of.fit.print=TRUE,
      sgp.config=NULL,
      sgp.config.drop.nonsequential.grade.progression.variables=TRUE,
      sgp.baseline.panel.years=NULL,
      sgp.baseline.config=NULL,
      parallel.config=NULL,
verbose.output=FALSE,
print.other.gp=FALSE,
get.cohort.data.info=FALSE,
...)
```

Arguments

<code>sgp_object</code>	An object of class SGP containing long formatted data in the code (from prepareSGP) slot.
<code>state</code>	Acronym indicating state associated with the data for access to embedded knot and boundaries, cutscores, CSEMs, and other state related assessment data.
<code>years</code>	A vector indicating year(s) in which to produce student growth percentiles and/or student growth projections/trajectories. If missing the function will use the data to infer the year(s) based upon the assumption of having at least three years of panel data for analyses.
<code>content_areas</code>	A vector indicating content area(s) in which to produce student growth percentiles and/or student growth projections/trajectories. If left missing the function will use the data to infer the content area(s) available for analyses.
<code>grades</code>	A vector indicating grades for which to calculate student growth percentiles and/or student growth projections/trajectories. If left missing the function will use the data to infer all the grade progressions for student growth percentile and student growth projections/trajectories analyses.
<code>sgp.percentiles</code>	Boolean variable indicating whether to calculate student growth percentiles. Defaults to TRUE.
<code>sgp.projections</code>	Boolean variable indicating whether to calculate student growth projections. Defaults to TRUE.
<code>sgp.projections.lagged</code>	Boolean variable indicating whether to calculate lagged student growth projections often used for growth to standard analyses. Defaults to TRUE.
<code>sgp.percentiles.baseline</code>	Boolean variable indicating whether to calculate baseline student growth percentiles and/or coefficient matrices. Defaults to FALSE.
<code>sgp.projections.baseline</code>	Boolean variable indicating whether to calculate baseline student growth projections. Defaults to FALSE.
<code>sgp.projections.lagged.baseline</code>	Boolean variable indicating whether to calculate lagged baseline student growth projections. Defaults to FALSE.

<code>sgp.percentiles.baseline.max.order</code>	Integer indicating the maximum order to calculate baseline student growth percentiles (regardless of maximum coefficient matrix order). Default is 3. To utilize the maximum matrix order, set to NULL.
<code>sgp.projections.baseline.max.order</code>	Integer indicating the maximum order to calculate baseline student growth projections (regardless of maximum coefficient matrix order). Default is 3. To utilize the maximum matrix order, set to NULL.
<code>sgp.projections.lagged.baseline.max.order</code>	Integer indicating the maximum order to calculate lagged baseline student growth projections (regardless of maximum coefficient matrix order). Default is 3. To utilize the maximum matrix order, set to NULL.
<code>sgp.minimum.default.panel.years</code>	Integer indicating the minimum number of panels years to use for default sgp analyses. Default value is 3 years of data.
<code>sgp.use.my.coefficient.matrices</code>	Argument, defaults to NULL indicating whether to use coefficient matrices to calculate student growth percentiles embedded in provided object of same name as those provided by the <code>sgp.labels</code> argument.
<code>simulate.sgps</code>	Boolean variable indicating whether to simulate SGP values for students based on test-specific Conditional Standard Errors of Measurement (CSEM). Test CSEM data must be available for simulation and included in <code>SGPstateData</code> . This argument must be set to TRUE for confidence interval construction. Defaults to TRUE.
<code>calculate.simex</code>	A character state acronymn or list including state/csem variable, simulation.iterations, lambda and extrapolation method. Defaults to NULL, no simex calculations performed.
<code>goodness.of.fit.print</code>	Boolean variable indicating whether to print out Goodness of Fit figures as PDF into a directory labeled Goodness of Fit. Defaults to TRUE.
<code>sgp.config</code>	If years, content_areas, and grades are missing, user can directly specify a list containing three vectors: <code>baseline.content_areas</code> , <code>baseline.panel.years</code> , and <code>baseline.grade.sequences</code> . This advanced option is helpful for analysis of non-traditional grade progressions and other special cases. See examples for use cases.
<code>sgp.config.drop.nonsequential.grade.progression.variables</code>	Boolean variable (defaults to TRUE) indicating whether non-sequential grade progression variables should be dropped when <code>sgp.config</code> is processed. For example, if a grade progression of <code>c(3,4,6)</code> is provided, the data configuration will assume (default is TRUE) that data for a missing year needs to be dropped prior to applying studentGrowthPercentiles or studentGrowthProjections to the data.
<code>sgp.baseline.panel.years</code>	A vector of years to be used for baseline coefficient matrix calculation. Default is to use most recent five years of data.

`sgp.baseline.config`

A list containing three vectors: `sgp.content.areas`, `sgp.panel.years`, `sgp.grade.sequences` indicating how baseline student growth percentile analyses are to be conducted. In almost all cases this value is calculated by default within the function but can be specified directly for advanced use cases. See source code for more detail on this configuration option.

`parallel.config`

A named list with, at a minimum, two elements indicating 1) the BACKEND package to be used for parallel computation and 2) the WORKERS list to specify the number of processors to be used in each major analysis. The BACKEND element can be set = to FOREACH or PARALLEL. Please consult the manuals and vignettes for information of these packages!

TYPE is a third element of the `parallel.config` list that provides necessary information when using FOREACH or PARALLEL packages as the backend. With BACKEND="FOREACH", the TYPE element specifies the flavor of 'foreach' backend. As of version 1.0-1.0, only "doParallel" is supported. If BACKEND = "PARALLEL", the `parallel` package will be used. This package combines deprecated parallel packages `snow` and `multicore`. Using the "snow" implementation of `parallel` the function will create a cluster object based on the TYPE element specified and the number of workers requested (see WORKERS list description below). The TYPE element indicates the users preferred cluster type (either "SOCK" for socket cluster or "MPI" for an OpenMPI cluster). If Windows is the operating system, this "snow" implementation must be use and the TYPE element must = "SOCK". Defaults are assigned based on operating system if TYPE is missing based on system OS. Unix/Mac OS defaults to the "multicore" to avoid worker node prescheduling and appears to be more efficient in these operating systems.

The WORKERS list must contain, at a minimum, a single number of processors (nodes) desired or available. If WORKERS is specified in this manner, then the same number of processors will be used for each analysis type (`sgp.percentiles`, `sgp.projections`, ... `sgp.projections.lagged.baseline`). Alternatively, the user may specify the numbers of processors used for each analysis. This allows for better memory management in systems that do not have enough RAM available per core. The choice of the number of cores is a balance between the number of processors available, the amount of RAM a system has and the size of the data (`sgp_object`). Each system will be different and will require some tailoring. One rule of thumb used by the authors is to allow for 4GB of memory per core used for running large state data. The SGP Demonstration (and data that size) requires more like 1-2GB per core. As an example, PERCENTILES=4 and PROJECTIONS=2 might be used on a quad core machine with 4 GB of RAM. This will use all 4 cores available for the `sgp.percentiles` analysis and 2 cores for the `sgp.projections` analysis (which requires more memory than available). The WORKERS list accepts these elements: PERCENTILES, PROJECTIONS (for both cohort and baseline referenced projections), LAGGED_PROJECTIONS (for both cohort and baseline referenced lagged projections), BASELINE_MATRICES (used to produce the baseline coefficient matrices when not available in `SGPstateData` - very computationally intensive), BASELINE_PERCENTILES (SGP calculation only when

baseline coefficient matrices have already been produced and are available - NOT very computationally intensive).

Alternatively, the name of an external CLUSTER.OBJECT (SOCK or MPI) set up by the user outside of the function can be used.

Example use cases are provided below.

verbose.output	A boolean argument (defaults to FALSE) indicating whether the function should output verbose diagnostic messages.
print.other.gp	A boolean argument (defaults to FALSE) indicating whether the function should output SGP of all orders.
get.cohort.data.info	A boolean argument (defaults to FALSE) indicating whether the a summary of all cohorts to be submitted to the studentGrowthPercentiles and studentGrowthProjections functions should be performed prior to analysis.
...	Arguments to be passed to studentGrowthPercentiles or studentGrowthProjections for finer control over SGP calculations. NOTE: arguments can only be passed to one lower level function at a time, and only student growth percentiles OR projections can be created but not both at the same time.

Value

Function returns a list containing the long data set in the Data slot as a `data.table` keyed using `VALID_CASE`, `CONTENT_AREA`, `YEAR`, `ID` and the student growth percentile and/or student growth projection/trajectory results in the SGP slot.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <vaniwaarden@colorado.edu>

See Also

[prepareSGP](#), [combineSGP](#)

Examples

```
## Not run:
## analyzeSGP is Step 2 of 5 of abcSGP
Demonstration_SGP <- sgpData_LONG
Demonstration_SGP <- prepareSGP(Demonstration_SGP)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP)

## Or (explicitly pass state argument)

Demonstration_SGP <- prepareSGP(sgpData_LONG)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP, state="DEMO")

###
### Example uses of the sgp.config argument
###
```

```

# Use only 3 years of Data, for grades 3 to 6
# and only perform analyses for most recent year (2012)

my.custom.config <- list(
  MATHEMATICS.2012_2013 = list(
    sgp.content.areas=rep("MATHEMATICS", 3), # Note, must be same length as sgp.panel.years
    sgp.panel.years=c('2010_2011', '2011_2012', '2012_2013'),
    sgp.grade.sequences=list(3:4, 3:5, 4:6)),
  READING.2012_2013 = list(
    sgp.content.areas=rep("READING", 3),
    sgp.panel.years=c('2010_2011', '2011_2012', '2012_2013'),
    sgp.grade.sequences=list(3:4, 3:5, 4:6)))

Demonstration_SGP <- prepareSGP(sgpData_LONG)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP,
  sgp.config=my.custom.config,
  sgp.percentiles.baseline = FALSE,
  sgp.projections.baseline = FALSE,
  sgp.projections.lagged.baseline = FALSE,
  simulate.sgps=FALSE)

## Another example sgp.config list:

# Use different CONTENT_AREA priors, and only 1 year of prior data
my.custom.config <- list(
  MATHEMATICS.2012_2013.READ_PRIOR = list(
    sgp.content.areas=c("READING", "MATHEMATICS"),
    sgp.panel.years=c('2011_2012', '2012_2013'),
    sgp.grade.sequences=list(3:4, 4:5, 5:6)),
  READING.2012_2013.MATH_PRIOR = list(
    sgp.content.areas=c("MATHEMATICS", "READING"),
    sgp.panel.years=c('2011_2012', '2012_2013'),
    sgp.grade.sequences=list(3:4, 4:5, 5:6)))

## An example showing multiple priors within a single year

Demonstration_SGP <- prepareSGP(sgpData_LONG)

DEMO.config <- list(
  READING.2011_2012 = list(
    sgp.content.areas=c("MATHEMATICS", "READING", "MATHEMATICS", "READING", "READING"),
    sgp.panel.years=c('2009_2010', '2009_2010', '2010_2011', '2010_2011', '2011_2012'),
    sgp.grade.sequences=list(c(3,3,4,4,5), c(4,4,5,5,6), c(5,5,6,6,7), c(6,6,7,7,8))),
  MATHEMATICS.2011_2012 = list(
    sgp.content.areas=c("READING", "MATHEMATICS", "READING", "MATHEMATICS", "MATHEMATICS"),
    sgp.panel.years=c('2009_2010', '2009_2010', '2010_2011', '2010_2011', '2011_2012'),
    sgp.grade.sequences=list(c(3,3,4,4,5), c(4,4,5,5,6), c(5,5,6,6,7), c(6,6,7,7,8)))

Demonstration_SGP <- analyzeSGP(Demonstration_SGP,
  sgp.config=DEMO.config,
  sgp.projections=FALSE,

```

```

sgp.projections.lagged=FALSE,
sgp.percentiles.baseline=FALSE,
sgp.projections.baseline=FALSE,
sgp.projections.lagged.baseline=FALSE,
sgp.config.drop.nonsequential.grade.progression.variables=FALSE)

###
### Example uses of the parallel.config argument
###

## Windows users must use a snow socket cluster:
# possibly a quad core machine with low RAM Memory
# 4 workers for percentiles, 2 workers for projections.
# Note the SOCK type cluster is used for single machines.

Demonstration_SGP <- prepareSGP(sgpData_LONG)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP,
parallel.config=list(
  BACKEND="PARALLEL", TYPE="SOCK",
  WORKERS=list(PERCENTILES=4,
    PROJECTIONS=2,
    LAGGED_PROJECTIONS=2,
    BASELINE_PERCENTILES=4))

## New parallel package - only available with R 2.13 or newer
# Note there are up to 16 workers, and MPI is used,
# suggesting this example is for a HPC cluster, possibly Windows OS.
...
parallel.config=list(
  BACKEND="PARALLEL", TYPE="MPI",
  WORKERS=list(PERCENTILES=16,
    PROJECTIONS=8,
    LAGGED_PROJECTIONS=6,
    BASELINE_PERCENTILES=12))
...

## FOREACH use cases:
...
parallel.config=list(
  BACKEND="FOREACH", TYPE="doParallel",
  WORKERS=3)
...

# NOTE: This list of parallel.config specifications is NOT exhaustive.
# See examples in analyzeSGP documentation for some others.0

###
### Advanced Example: restrict years, recalculate baseline SGP
### coefficient matrices, and use parallel processing
###

```

```

# Remove existing DEMO baseline coefficient matrices from
# the SGPstateData object so that new ones will be computed.

SGPstateData$DEMO$Baseline_splineMatrix <- NULL

# set up a customized sgp.config list

. . .

# set up a customized sgp.baseline.config list

. . .

# to be completed

## End(Not run)

```

baselineSGP	<i>Analyze student data to produce student growth percentiles and coefficient matrices from a baseline (i.e. multiple cohort) norm group</i>
-------------	--

Description

Utility function/exemplar used to produce student growth percentiles using long formatted data like that provided by [prepareSGP](#). Used as part of [analyzeSGP](#) for baseline referenced student growth percentile analyses.

Usage

```

baselineSGP(sgp_object,
            state=NULL,
            years=NULL,
            content_areas=NULL,
            grades=NULL,
            sgp.config=NULL,
            sgp.baseline.config=NULL,
            sgp.baseline.panel.years=NULL,
            sgp.percentiles.baseline.max.order=3,
            return.matrices.only=FALSE,
            calculate.baseline.sgps=TRUE,
            goodness.of.fit.print=TRUE,
            ...)

```

Arguments

sgp_object	An object of class SGP containing long formatted data in the Data (from prepareSGP) slot.
------------	--

<code>state</code>	Acronym indicating state associated with the data for access to embedded knot and boundaries.
<code>years</code>	A vector indicating year(s) in which to produce baseline referenced student growth percentiles.
<code>content_areas</code>	A vector indicating content area in which to produce baseline referenced student growth percentiles.
<code>grades</code>	A vector indicating which grades to calculate baseline referenced student growth percentiles.
<code>sgp.config</code>	If years, content_areas, and grades are missing, user can directly specify a list containing three vectors: <code>baseline.content_areas</code> , <code>baseline.panel.years</code> , and <code>baseline.grade.sequences</code> . This advanced option is helpful for analysis of non-traditional grade progressions and other special cases. See analyzeSGP for use cases.
<code>sgp.baseline.config</code>	A list containing three vectors: <code>sgp.content_areas</code> , <code>sgp.panel.years</code> , <code>sgp.grade.sequences</code> indicating how baseline student growth percentile analyses are to be conducted. In most cases this value will be calculated by default within the function but can be specified directly for advanced use cases. See source code for more detail on this configuration option.
<code>sgp.baseline.panel.years</code>	A character vector indicating the years to be used for the calculation of baseline SGPs. Default is to use most recent five years of data.
<code>sgp.percentiles.baseline.max.order</code>	Integer indicating the maximum order to calculate baseline student growth percentiles (regardless of maximum coefficient matrix order). Default is 3. To utilize the maximum matrix order, set to NULL.
<code>return.matrices.only</code>	Boolean variable indicating whether the function will only return baseline referenced coefficient matrices. Defaults to FALSE.
<code>calculate.baseline.sgps</code>	Boolean variable indicating whether the function will calculate baseline referenced student growth percentiles from baseline referenced coefficient matrices. Defaults to TRUE.
<code>goodness.of.fit.print</code>	Boolean variable indicating whether the function will export goodness of fit plots if baseline referenced student growth percentiles are calculated. Defaults to TRUE.
<code>...</code>	Arguments to be passed internally to studentGrowthPercentiles for finer control over SGP calculations.

Value

If `return.matrices.only` is set to TRUE function returns a list containing the baseline referenced coefficient matrices. Otherwise function returns the SGP object provided with the `sgp_object` argument with the baseline referenced coefficient matrices, growth percentiles, etc. embedded.

Author(s)

Adam Van Iwaarden <Vaniwaarden@colorado.edu>, Ben Domingue <ben.domingue@gmail.com>
and Damian W. Betebenner <dbetebenner@nciea.org>

See Also

[prepareSGP](#), [analyzeSGP](#), [combineSGP](#)

Examples

```
## Not run:
## Calculate baseline referenced SGPs (using coefficient matrices embedded in SGPstateData)

Demonstration_SGP <- prepareSGP(sgpData_LONG)
Demonstration_SGP <- baselineSGP(Demonstration_SGP)

## Calculate baseline referenced coefficient matrices

SGPstateData[["DEMO"]][["Baseline_splineMatrix"]] <- NULL
Demonstration_SGP <- prepareSGP(sgpData_LONG)
DEMO_Baseline_Matrices <- baselineSGP(Demonstration_SGP, return.matrices.only=TRUE, calculate.baseline.sgps=FALSE)

## Calculate baseline referenced coefficient matrices and baseline referenced SGPs with 4 years of data

SGPstateData[["DEMO"]][["Baseline_splineMatrix"]] <- NULL

sgpData_LONG_4_YEAR <- subset(sgpData_LONG, YEAR!="2012_2013")

Demonstration_SGP <- prepareSGP(sgpData_LONG_4_YEAR)
Demonstration_SGP <- baselineSGP(Demonstration_SGP)

## End(Not run)
```

bubblePlot

Core bubblePlot function for SGP

Description

Function to create bubble plots associated with student growth percentile results. The function is adaptable to many representation but is used in conjunction with results derived from [summarizeSGP](#) to represent summary level results of growth against achievement (usually, median student growth percentile against percentage at/above proficient). The function has MANY options and users are advised to read this documentation thoroughly as well as investigate the source code for the function itself to see what the many different representations that are possible. The function has the ability to produce interactive data tips using the pdf2 package available on R-Forge. This package is NOT installed as part of the SGP package and must be installed separately from the package to take advantage of this functionality of the bubblePlot function. To install pdf2 from the R prompt type: `install.packages("pdf2", repos="http://R-Forge.R-project.org")`

Usage

```

bubblePlot(
  bubble_plot_data.X,
  bubble_plot_data.Y,
  bubble_plot_data.SUBSET=NULL,
  bubble_plot_data.INDICATE=NULL,
    bubble_plot_data.BUBBLE_CENTER_LABEL=NULL,
  bubble_plot_data.SIZE,
  bubble_plot_data.LEVELS=NULL,
  bubble_plot_data.BUBBLE_TIPS_LINES,
  bubble_plot_labels.X=c("Growth", "Median Student Growth Percentile"),
  bubble_plot_labels.Y=c("Achievement", "Percent at/above Proficient"),
  bubble_plot_labels.SIZE=c(50, 100, 500, 1000),
  bubble_plot_labels.LEVELS=NULL,
  bubble_plot_labels.BUBBLE_TIPS_LINES=list("Median SGP (Count)",
    "Percent at/above Proficient"),
  bubble_plot_labels.BUBBLE_TITLES,
  bubble_plot_titles.MAIN="Growth and Achievement",
  bubble_plot_titles.SUB1="State School Performance",
  bubble_plot_titles.SUB2="Growth & Current Achievement",
  bubble_plot_titles.LEGEND1="School Size",
  bubble_plot_titles.LEGEND2_P1=NULL,
  bubble_plot_titles.LEGEND2_P2=NULL,
  bubble_plot_configs.BUBBLE_MIN_MAX=c(0.03, 0.03),
  bubble_plot_configs.BUBBLE_X_TICKS=seq(0,100,10),
  bubble_plot_configs.BUBBLE_X_TICKS_SIZE=c(rep(0.6, 5), 1, rep(0.6, 5)),
    bubble_plot_configs.BUBBLE_X_BANDS=NULL,
    bubble_plot_configs.BUBBLE_X_BAND_LABELS=NULL,
  bubble_plot_configs.BUBBLE_Y_TICKS=seq(0,100,10),
  bubble_plot_configs.BUBBLE_Y_TICKS_SIZE=rep(0.6, 11),
    bubble_plot_configs.BUBBLE_Y_BANDS=NULL,
    bubble_plot_configs.BUBBLE_Y_BAND_LABELS=NULL,
  bubble_plot_configs.BUBBLE_SUBSET_INCREASE=0,
  bubble_plot_configs.BUBBLE_SUBSET_ALPHA=list(Transparent=0.3, Opaque=0.95),
  bubble_plot_configs.BUBBLE_COLOR="deeppink2",
    bubble_plot_configs.BUBBLE_COLOR_GRADIENT_REVERSE=FALSE,
  bubble_plot_configs.BUBBLE_TIPS=TRUE,
  bubble_plot_configs.BUBBLE_PLOT_DEVICE="PDF",
  bubble_plot_configs.BUBBLE_PLOT_FORMAT="print",
  bubble_plot_configs.BUBBLE_PLOT_LEGEND=FALSE,
  bubble_plot_configs.BUBBLE_PLOT_TITLE=TRUE,
  bubble_plot_configs.BUBBLE_PLOT_BACKGROUND_LABELS=c("Growth", "Achievement"),
  bubble_plot_configs.BUBBLE_PLOT_EXTRAS=NULL,
    bubble_plot_configs.BUBBLE_PLOT_DIMENSION=NULL, ## List of WIDTH and HEIGHT
  bubble_plot_configs.BUBBLE_PLOT_NAME="bubblePlot.pdf",
  bubble_plot_configs.BUBBLE_PLOT_PATH=paste("Figures", sep=""),
  bubble_plot_pdftk.CREATE_CATALOG=FALSE)

```

Arguments`bubble_plot_data.X`

A vector of X coordinates for the bubbles to be plotted.

`bubble_plot_data.Y`

A vector of Y coordinates for the bubbles to be plotted.

`bubble_plot_data.SUBSET`

A boolean vector indicating a subset of the bubbles to be highlighted and plotted. When `BUBBLE_TIPS` are indicated, only subsetted bubbles will show bubble tips. To further accentuate highlight bubbles, their radius can be altered using the `bubble_plot_configs.BUBBLE_SUBSET_INCREASE` argument. Default value is `bubble_plot_data.SUBSET=NULL`.

`bubble_plot_data.INDICATE`

A boolean vector indicating whether to attach a label to further highlight in a manner suitable for printing. Usually done for few bubbles. Default value is `bubble_plot_data.INDICATE=NULL`.

`bubble_plot_data.BUBBLE_CENTER_LABEL`

A character vector to label the interior of the bubbles with. Usually a vector of singleton characters. Default value is `bubble_plot_data.BUBBLE_CENTER_LABEL=NULL`.

`bubble_plot_data.SIZE`

A vector indicating the size of each of the bubbles plotted.

`bubble_plot_data.LEVELS`

A vector (usually a factor) indicating categories to which the bubbles belong. Default value is `bubble_plot_data.LEVELS=NULL`.

`bubble_plot_data.BUBBLE_TIPS_LINES`

A list of arbitrary length indicating the different values supplied when bubble tips are requested. Default value is `bubble_plot_data.BUBBLE_TIPS_LINES=list(paste(MEDIAN_SGP`

`bubble_plot_labels.X`

A vector of length 2 where the 1st component is, usually a one word summary for the axis (e.g., Growth) and the 2nd component of the vector is a label for the axis (e.g., Median Student Growth Percentile). Default value is `bubble_plot_labels.X=c('Growth', 'Median Student Growth Percentile')`.

`bubble_plot_labels.Y`

A vector of length 2 where the 1st component is, usually a one word summary for the axis (e.g., Achievement) and the 2nd component of the vector is a label for the axis (e.g., Percent at/above Proficient). Default value is `bubble_plot_labels.Y=c('Achievement`

`bubble_plot_labels.SIZE`

A vector of quantities giving breaking points for the size bubbles indicated in the legend of the plot. Default value is `bubble_plot_labels.SIZE=c(50, 100, 500, 1000)`.

`bubble_plot_labels.LEVELS`

A vector of quantities giving level labels associated with `bubble_plot_data.LEVELS`. These labels will appear in the right legend of the plot. Default value is `bubble_plot_labels.LEVELS=NU`

`bubble_plot_labels.BUBBLE_TIPS_LINES`

A list of labels that appear in the mouse over data tips. Should be of same length as the list from `bubble_plot_data.BUBBLE_TIPS_LINES`. Default value is `bubble_plot_labels.BUBBLE_TIPS_LINES=list('Median SGP (Count)', 'Percent at/above`

`bubble_plot_labels.BUBBLE_TITLES`
 A character vector with of the same length as number of points plotted indicated what name should appear on each mouse over bubble tip (e.g., the school name associated with the bubble. Default value is `bubble_plot_labels.BUBBLE_TITLES=SCHOOL_NAME`,

`bubble_plot_titles.MAIN`
 The main title of the bubble plot. Default value is `bubble_plot_titles.MAIN='Growth and Achieveme`

`bubble_plot_titles.SUB1`
 The right upper title of the bubble plot. Default value is `bubble_plot_titles.SUB1='State School Pe`

`bubble_plot_titles.SUB2`
 The right lower title of the bubble plot. Default value is `bubble_plot_titles.SUB2='Growth & Curren`

`bubble_plot_titles.LEGEND1`
 The title of the upper legend to the right of the bubble plot. Default value is `bubble_plot_titles.LEGEND1='School Size'`.

`bubble_plot_titles.LEGEND2_P1`
 The 1st line of the title of the lower legend of the bubble plot. Default value is `bubble_plot_titles.LEGEND2_P1=NULL`.

`bubble_plot_titles.LEGEND2_P2`
 The 2nd line of the title of the lower legend of the bubble plot. Default value is `bubble_plot_titles.LEGEND2_P2=NULL`.

`bubble_plot_configs.BUBBLE_MIN_MAX`
 A vector of length two indicating min and max values for the bubbles in inches. Default value is `bubble_plot_configs.BUBBLE_MIN_MAX=c(0.03, 0.03)`.

`bubble_plot_configs.BUBBLE_X_TICKS`
 A vector indicating what x coordinates to display for the x axis of the bubble plot. Default value is `bubble_plot_configs.BUBBLE_X_TICKS=seq(0,100,10)`.

`bubble_plot_configs.BUBBLE_X_TICKS_SIZE`
 A vector, the same length as `bubble_plot_configs.BUBBLE_X_TICKS` indicating the character expansion (i.e., `cex`) associated with the characters. Default value is `bubble_plot_configs.BUBBLE_X_TICKS_SIZE=c(rep(0.6, 5), 1, rep(0.6, 5))`.

`bubble_plot_configs.BUBBLE_X_BANDS`
 A vector of cutpoints used to separate the plot background into horizontal rectangular regions.

`bubble_plot_configs.BUBBLE_X_BAND_LABELS`
 A character vector of labels used to labels the rectangles produced with `bubble_plot_configs.BUBBLE_X`

`bubble_plot_configs.BUBBLE_Y_TICKS`
 A vector indicating what x coordinates to display for the x axis of the bubble plot. Default value is `bubble_plot_configs.BUBBLE_Y_TICKS=seq(0,100,10)`.

`bubble_plot_configs.BUBBLE_Y_TICKS_SIZE`
 A vector, the same length as `bubble_plot_configs.BUBBLE_Y_TICKS` indicating the character expansion (i.e., `cex`) associated with the characters. Default value is `bubble_plot_configs.BUBBLE_Y_TICKS_SIZE=rep(0.6, 11)`.

`bubble_plot_configs.BUBBLE_Y_BANDS`
 A vector of cutpoints used to separate the plot background into horizontal rectangular regions.

`bubble_plot_configs.BUBBLE_Y_BAND_LABELS`
 A character vector of labels used to labels the rectangles produced with `bubble_plot_configs.BUBBLE_X`

```

bubble_plot_configs.BUBBLE_SUBSET_INCREASE
    Default value is bubble_plot_configs.BUBBLE_SUBSET_INCREASE=0.
bubble_plot_configs.BUBBLE_SUBSET_ALPHA
    Default value is bubble_plot_configs.BUBBLE_SUBSET_ALPHA=list(Transparent=0.3, Opaque=0.
bubble_plot_configs.BUBBLE_COLOR
    Default value is bubble_plot_configs.BUBBLE_COLOR="deeppink2".
bubble_plot_configs.BUBBLE_COLOR_GRADIENT_REVERSE
    Boolean argument (defaults to FALSE) indicating whether to reverse color gra-
    dient associated with bubbles.
bubble_plot_configs.BUBBLE_TIPS
    Default value is bubble_plot_configs.BUBBLE_TIPS=TRUE.
bubble_plot_configs.BUBBLE_PLOT_DEVICE
    Default value is bubble_plot_configs.BUBBLE_PLOT_DEVICE='PDF'.
bubble_plot_configs.BUBBLE_PLOT_FORMAT
    Default value is bubble_plot_configs.BUBBLE_PLOT_FORMAT='print'.
bubble_plot_configs.BUBBLE_PLOT_LEGEND
    Default value is bubble_plot_configs.BUBBLE_PLOT_LEGEND=FALSE.
bubble_plot_configs.BUBBLE_PLOT_TITLE
    Default value is bubble_plot_configs.BUBBLE_PLOT_TITLE=TRUE.
bubble_plot_configs.BUBBLE_PLOT_BACKGROUND_LABELS
    Default value is bubble_plot_configs.BUBBLE_PLOT_BACKGROUND_LABELS=c('Growth', 'Achievem
bubble_plot_configs.BUBBLE_PLOT_EXTRAS
    Default value is bubble_plot_configs.BUBBLE_PLOT_EXTRAS=NULL.
bubble_plot_configs.BUBBLE_PLOT_DIMENSION
    Default value is bubble_plot_configs.BUBBLE_PLOT_DIMENSION=NULL.
bubble_plot_configs.BUBBLE_PLOT_NAME
    Default value is bubble_plot_configs.BUBBLE_PLOT_NAME='bubblePlot.pdf'.
bubble_plot_configs.BUBBLE_PLOT_PATH
    Default value is bubble_plot_configs.BUBBLE_PLOT_PATH=paste('Figures', sep='').
bubble_plot_pdftk.CREATE_CATALOG
    Default value is bubble_plot_pdftk.CREATE_CATALOG=FALSE.

```

Details

Typical use of the function is as part of visualizeSGP function. However, function can be used more generically for diverse plots showing many dimensions of data simultaneously.

Value

Function creates a bubble chart and writes the result as a PDF to bubble_plot_configs.BUBBLE_PATH.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

bubblePlot_Styles	<i>bubblePlot_Styles providing various uses of the core bubblePlot function</i>
-------------------	---

Description

Function includes a number of "styles" associated with `bubblePlot` to create bubble plots depicting a variety of relationships often of interest to stakeholders. The `bubblePlot` function itself is adaptable to many representations but is most often used in conjunction with results derived from `summarizeSGP` to represent summary level results of growth against achievement (usually, median student growth percentile against percentage at/above proficient).

Usage

```
bubblePlot_Styles(
  sgp_object,
  state,
  bPlot.years=NULL,
  bPlot.content_areas=NULL,
  bPlot.districts=NULL,
  bPlot.schools=NULL,
  bPlot.instructors=NULL,
  bPlot.styles=c(1),
  bPlot.levels=NULL,
  bPlot.level.cuts=NULL,
  bPlot.full.academic.year=TRUE,
  bPlot.minimum.n=10,
  bPlot.anonymize=FALSE,
  bPlot.prior.achievement=TRUE,
  bPlot.draft=FALSE,
  bPlot.demo=FALSE,
  bPlot.format="print",
  bPlot.folder="Visualizations/bubblePlots")
```

Arguments

sgp_object	An object of class SGP containing long formatted data in the Data slot that will be used for the production of student growth and achievement plots and system growth and achievement plots, summary data from <code>summarizeSGP</code> in the Summary slot for bubble plots.
state	Acronym indicating state associated with the summaries for access to assessment program information embedded in <code>SGPstateData</code> .
bPlot.years	A vector indicating year(s) in which to produce <code>bubblePlots</code> using data available in <code>sgp_object</code> . If missing the function will use the last year available in the data to produce <code>bubblePlots</code> .

bPlot.content_areas	A vector indicating content area(s) to produce bubblePlots using data available in <code>sgp_object</code> . If missing the function will produce plots for all available content areas provided in the data.
bPlot.districts	A vector indicating districts to produce bubblePlots using data available in <code>sgp_object</code> . If missing the function will produce plots for all available districts provided in the data where districts represent a relevant unit to be represented by the specific bubblePlot style.
bPlot.schools	A vector indicating schools to produce bubblePlots using data available in <code>sgp_object</code> . If missing the function will produce plots for all available schools provided in the data where districts represent a relevant unit to be represented by the specific bubblePlot style.
bPlot.instructors	A vector indicating instructors to produce bubblePlots using data available in <code>sgp_object</code> . If missing the function will produce plots for all available instructors provided in the data where schools and districts represent relevant units to be represented by the specific bubblePlot style.
bPlot.styles	A vector of positive integers indicating the types of bubblePlots to produce using data available in <code>sgp_object</code> . See associated documentation for example plots.
bPlot.levels	A character vector of levels to be used to color bubbles in the bubblePlot . See associated documentation for example plots.
bPlot.level.cuts	A vector of cuts to be used to distinguish levels used to color bubbles in the bubblePlot . See associated documentation for example plots.
bPlot.full.academic.year	A boolean argument (defaults to TRUE) indicating whether bubblePlots should use full academic year results if available.
bPlot.minimum.n	A positive integer (defaults to 10) indicating the minimum size for summary values to be displayed in the bubblePlots .
bPlot.anonymize	A boolean argument (defaults to FALSE) indicating whether to anonymize bubblePlots school and district names that appear in the plots and data tips of the plots. For student level anonymization, the function utilizes the <code>randomNames</code> package to produce gender and ethnic correct names based upon gender and ethnicity codes available in <code>sgp_object@Data</code> .
bPlot.prior.achievement	A boolean argument (defaults to TRUE) indicating whether to produce bubblePlots using prior achievement as well as current achievement as the vertical dimension of the bubblePlot .
bPlot.draft	A boolean argument (defaults to FALSE) indicating whether to put an indicator on the chart noting that the results are draft and to not distribute.
bPlot.demo	A boolean argument (defaults to FALSE) indicating whether to produce demo student level plots (styles 150 and/or 153) for instructors.

bPlot.format	Either "print" or "presentation" indicating whether to optimize the plot for print form (light background) or presentation form (dark background).
bPlot.folder	Character vector indicating where bubblePlots should be placed. Default folder is "Visualizations/bubblePlots".

Details

bubblePlot_Styles is an omnibus function containing a number of applications of the bubblePlot function with data output from SGP analyses. As added styles are constructed they will be added to the function allowing user to construct plots of their liking. User wishing to participate and provide or suggest specific styles of their own should contact the package maintainer. Styles representing summary level data (e.g., growth and achievement for schools) are currently assigned numbers from 1 to 99 and styles representing individual level data (e.g., growth and achievement for students within a single grade) are currently assigned numbers from 100 to 199.

- 1: Growth and achievement bubblePlots for the entire state with bubbles representing schools.
- 100: Growth and achievement bubblePlots by grade for students in selected schools and districts.

Value

Function produces numerous PDF bubble charts in the styles specified by the function.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <vaniwaarden@colorado.edu>

capwords	<i>Function for converting all caps to mixed case. Useful in data cleaning.</i>
----------	---

Description

The function capwords converts characters to mixed case character as intelligently as possible.

Usage

```
capwords(x,  
  special.words = c("ELA", "II", "III", "IV", "EMH", "HS", "MS", "ES", "SES", "IEP", "ELL", "MAD", "
```

Arguments

x	A character string to be converted to mixed case.
special.words	A character vector (see default above), specifying words to not convert to mixed case.

Value

Returns a mixed case character string.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

Examples

```
## Not run:
capwords("TEST") ## Test
capwords("TEST1 TEST2") ## Test1 Test2
capwords("O'NEIL") ## O'Neil
capwords("JOHN'S") ## John's

## Use sapply for converting character vectors

test.vector <- paste("TEST", 1:10, sep="")
sapply(test.vector, capwords)

## End(Not run)
```

combineSGP

Combine student data and SGP results

Description

Utility function/exemplar used to merge student long data with results from student growth percentiles and/or student growth projections calculations. Default values of this function are designed to be used following use of other utility functions: [prepareSGP](#) and [analyzeSGP](#). Function is integrate with cutscores embedded in [SGPstateData](#) to calculate growth-to-standard targets associated with catch-up/keep-up to proficient status and move-up/stay-up to advanced status.

Usage

```
combineSGP(sgp_object,
state=NULL,
years=NULL,
content_areas=NULL,
sgp.percentiles=TRUE,
sgp.percentiles.baseline=TRUE,
sgp.projections.lagged=TRUE,
sgp.projections.lagged.baseline=TRUE,
max.sgp.target.years.forward=3,
update.all.years=FALSE)
```

Arguments

<code>sgp_object</code>	A list containing both Student (from <code>prepareSGP</code>) and SGP (from <code>analyzeSGP</code>) slots.
<code>state</code>	Acronym for which state is to be used for the lagged projections and growth to standard analyses. Function will try to guess state name from passed <code>sgp_object</code> is missing.
<code>years</code>	A vector of years indicating years of data to merge with <code>@Data</code> . If missing, merge will use all available years of student growth percentile results.
<code>content_areas</code>	A vector of content areas indicating content areas of student growth percentile data to merge with <code>@Data</code> . If missing, merge will use all available content areas of student growth percentile results.
<code>sgp.percentiles</code>	A boolean variable indicating whether to combine student growth percentiles.
<code>sgp.percentiles.baseline</code>	A boolean variable indicating whether to combine baseline student growth percentiles.
<code>sgp.projections.lagged</code>	A boolean variable indicating whether to combine lagged student growth projections and calculate catch-up/keep-up and move-up/stay-up values.
<code>sgp.projections.lagged.baseline</code>	A boolean variable indicating whether to combine lagged baseline student growth projections and calculate catch-up/keep-up move-up/stay-up values.
<code>max.sgp.target.years.forward</code>	A integer indicating the number of years forward from the lagged (last year's) score to project forward for growth to standard calculations. Default is 3 years from the present, 4 years from the lagged year, which is the standard in most growth to standard calculations used by state departments of education.
<code>update.all.years</code>	A boolean argument defaulting to FALSE indicating whether <code>combineSGP</code> should delete previously merged variables calculated in analyzeSGP and re-merge all available data.

Value

Function returns a list containing the input long data set in the Student slot as a `data.table` keyed using `VALID_CASE`, `CONTENT_AREA`, `YEAR`, `ID` merged with student growth percentiles and/or lagged projection targets and catch-up/keep-up and move-up/stay-up status.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <vaniwaarden@colorado.edu>

See Also

[prepareSGP](#), [analyzeSGP](#)

Examples

```
## Not run:
## combineSGP is Step 3 of 5 of abcSGP
Demonstration_SGP <- sgpData_LONG
Demonstration_SGP <- prepareSGP(Demonstration_SGP)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP)
Demonstration_SGP <- combineSGP(Demonstration_SGP)

## End(Not run)
```

`createKnotsBoundaries` *Function to create Knots and Boundaries from supplied data in LONG format.*

Description

The function `createKnotsBoundaries` creates Knots, Boundaries and Loss/Hoss scores for subsequent use and embedding in `SGPstateData`.

Usage

```
createKnotsBoundaries(tmp.data,
  knot.cut.percentiles=c(0.2,0.4,0.6,0.8))
```

Arguments

<code>tmp.data</code>	Data supplied to function in LONG format. See <code>sgpData_LONG</code> for exemplar. <code>tmp.data</code> must contain, at least, variables 'VALID_CASE', 'CONTENT_AREA', 'SCALE_SCORE', 'GRADE'
<code>knot.cut.percentiles</code>	A numeric vector of quantiles of 'SCALE_SCORE' to be used as the cut points. Default is to use the 20th, 40th, 60th, and 80th percentiles (i.e., <code>c(0.2,0.4,0.6,0.8)</code>).

Value

Returns a list containing Knots, Boundaries and Loss/Hoss scores.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

Examples

```
## Not run:
DEMO_Knots_Boundaries <- createKnotsBoundaries(sgpData_LONG)

## End(Not run)
```

gofSGP

*Function for producing goodness of fit plots using existing SGP object***Description**

gofSGP creates goodness-of-fit plots in either PDF or PNG for showing SGP distribution by prior achievement level and prior scale score decile. These plots expand upon the plots currently produced with the [studentGrowthPercentiles](#) function.

Usage

```
gofSGP(
  sgp_object,
  state=NULL,
  years=NULL,
  content_areas=NULL,
  content_areas_prior=NULL,
  grades=NULL,
  use.sgp="SGP",
  output.format="PDF",
  color.scale="reds.and.blues")
```

Arguments

sgp_object	The SGP object from which the goodness-of-fit data will be used.
state	The 'state' for the sgp_object. Derive from sgp_object name if not explicitly supplied.
years	The years that goodness-of-fit plots are requested. Default is to use all years available.
content_areas	The content area(s) that goodness-of-fit plots are requested. Default is to use all content areas available.
content_areas_prior	The content area(s) of the prior year which growth by achievement level is being produced that goodness-of-fit plots are requested. Default is to use all content areas available.
grades	The grade(s) that goodness-of-fit plots are requested. Default is to use all grade available.
use.sgp	Character vectors (defaults to 'SGP') indicating what student growth percentile variable to calculate goodness-of-fit plots for.
output.format	Character vectors (defaults to 'PDF') indicating what driver to use to output results. Options currently include 'PDF' and 'PNG'.
color.scale	Character vectors (defaults to 'red') indicating what color palette to use for creating percentile distribution table.

Value

Returns output ('PDF' or 'PNG') associated with goodness-of-fit analyses.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

See Also

[studentGrowthPercentiles](#)

Examples

```
## Not run:  
Demonstration_SGP <- abcSGP(sgpData_LONG)  
gofSGP(Demonstration_SGP)  
  
## End(Not run)
```

growthAchievementPlot *growthAchievementPlot for SGP*

Description

Function to create growth and achievement plots depicting system level results associated with student growth percentile results. The charts show, simultaneously, norm- and criterion-referenced student achievement (i.e., status) as well as norm- and criterion-referenced student growth. These charts are those shown on the cover of the December 2009 Issue of *Educational Measurement: Issues and Practice*. See Betebenner (2009) and Betebenner (2012) for more details

Usage

```
growthAchievementPlot(  
  gaPlot.sgp_object,  
  gaPlot.students,  
  gaPlot.percentile_trajectories,  
  gaPlot.achievement_percentiles=c(.01, seq(.05, .95, by=.05), .99),  
  gaPlot.show.scale.transformations=TRUE,  
  gaPlot.grade_range,  
  gaPlot.max.order.for.progression=NULL,  
  state,  
  content_area,  
  year,  
  format="print",  
  baseline=FALSE,  
  output.format="PDF",  
  output.folder,  
  assessment.name)
```

Arguments

<code>gaPlot.sgp_object</code>	The <code>sgp_object</code> containing system information for constructing the growth and achievement plot. Object is calculated using <code>abcSGP</code> or (at least) <code>prepareSGP</code> followed by <code>analyzeSGP</code> .
<code>gaPlot.students</code>	Either NULL (the default) or a list of student IDs for whom one wishes to generate growth and achievement plots.
<code>gaPlot.percentile_trajectories</code>	A vector indicating the growth percentile trajectories to be depicted on the plot. If missing, the percentile trajectories will be the trajectories associated with the state supplied. If no state is supplied, the percentile trajectories will be 10, 35, 50, 65, 90.
<code>gaPlot.achievement_percentiles</code>	A vector of percentiles that achievement (i.e., status) percentiles will be depicted across the range of grades.
<code>gaPlot.show.scale.transformations</code>	A boolean arguments (defaults to TRUE) indicating whether to show the scale as a vertical axis if a scale transformation is applied to the supplied data.
<code>gaPlot.grade_range</code>	The grade range for which to demonstrate plot. If missing, function uses supplied state to derive grade range.
<code>gaPlot.max.order.for.progression</code>	The maximum coefficient matrix order to use for each progression. Default is NULL which utilizes the maximum order available with the coefficient matrices.
<code>state</code>	A two letter acronym for the state associated with the supplied data.
<code>content_area</code>	A character vector indicating the content area to produce the growth and achievement chart. Note that the supplied content area must match that used in the internal labelling of content area for the <code>sgp_object</code> .
<code>year</code>	The year for which to produce the growth and achievement plots.
<code>format</code>	Either "print" (the default) for light background or "presentation" for dark background slides.
<code>baseline</code>	Boolean variable (defaults to FALSE) indicating whether to create percentile trajectories using baseline referenced coefficient matrices.
<code>output.format</code>	A character vector indicating which output format to use. Currently support 'PDF' (the default) and 'PNG'. The call to <code>growthAchievementPlot</code> in <code>visualizeSGP</code> outputs in both formats.
<code>output.folder</code>	A character vector indicating where to put the produced growth and achievement plot.
<code>assessment.name</code>	A character vector indicating the <code>assessment.name</code> . If missing, the supplied state is used to determine the <code>assessment.name</code> .

Details

Typical use of the function is as part of `visualizeSGP` function. However, function can be used to produce growth and achievement charts.

Value

Function creates a growth and achievement chart and writes the result as a PDF to `output` folder.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

References

- Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.
- Betebenner, D. W. (2009). Norm- and criterion-referenced student growth. *Educational Measurement: Issues and Practice*, 28(4):42-51.

outputSGP

Output student data and SGP results for a variety of purposes

Description

Utility function used to export student data and SGP results for a variety of purposes. Current functionality exports data in wide format for data visualization purposes. See source code for detailed functionality.

Usage

```
outputSGP(sgp_object,
state=NULL,
  output.type=c("LONG_Data", "WIDE_Data", "INSTRUCTOR_Data"),
  baseline.sgps=FALSE,
  outputSGP_SUMMARY.years=NULL,
  outputSGP_SUMMARY.content_areas=NULL,
  outputSGP_INDIVIDUAL.years=NULL,
  outputSGP_INDIVIDUAL.content_areas=NULL,
  outputSGP.anonymize=FALSE,
  outputSGP.student.groups=NULL,
  outputSGP.directory="Data",
outputSGP.translate.names=TRUE,
outputSGP.projection.years.for.target=3)
```

Arguments

- | | |
|------------|--|
| sgp_object | An object of class SGP containing data to be exported. |
| state | Acronym for which state is to be used for the lagged projections and growth to standard analyses. Function will try to guess state name from passed sgp_object is missing. |

<code>years</code>	A vector indicating the years to be included in the output. Default is to use all years.
<code>content_areas</code>	A vector indicating the content areas to be included in the output. Default is to use all content areas.
<code>output.type</code>	A character vector indicating what output type is requested. Currently <code>LONG_Data</code> , <code>WIDE_Data</code> , <code>INSTRUCTOR_Data</code> and <code>SchoolView</code> are supported modes of output. <code>LONG_Data</code> exports the contents of the <code>@Data</code> slot in a pipe delimited format. <code>WIDE_Data</code> exports a reshaped version of the <code>@Data</code> slot where each row is a unique student record. <code>INSTRUCTOR_Data</code> uses the <code>@Data_Supplementary\$INSTRUCTOR_NUMBER</code> table to export a long student by instructor number table. <code>SchoolView</code> exports tables used for representation in <code>SchoolView</code> . The default exports <code>LONG_Data</code> and <code>WIDE_Data</code> .
<code>baseline.sgps</code>	Boolean vector indicating whether to output baseline SGPs for cohorts referenced SGPs.
<code>outputSGP_SUMMARY.years</code>	A vector indicating the year to be used for output file construction for summary tables.
<code>outputSGP_SUMMARY.content_areas</code>	A vector indicating the content areas to be used for output file construction for summary tables.
<code>outputSGP_INDIVIDUAL.years</code>	A vector indicating the year to be used for output file construction for individual level file.
<code>outputSGP_INDIVIDUAL.content_areas</code>	A vector indicating the content areas to be used for output file construction for individual level file.
<code>outputSGP.anonymize</code>	A boolean variable indicating whether to anonymize output files.
<code>outputSGP.student.groups</code>	A list of variables to be used for student groups in individual and summary tables.
<code>outputSGP.directory</code>	A file path indicating where to save output files. Defaults to <code>Data</code> .
<code>outputSGP.translate.names</code>	A boolean argument, defaults to <code>TRUE</code> , indicating whether data output should refer to <code>'names.provided'</code> or <code>'names.sgp'</code> in <code>@Names</code> slot of supplied SGP object.
<code>outputSGP.projection.years.for.target</code>	An integer argument indicating what projection to supply with regard to the number of years projected forward.

Value

Function returns an R object with data to be exported per user's desire (e.g., as a pipe delimited file).

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

See Also

[abcSGP](#), [prepareSGP](#), [analyzeSGP](#), [combineSGP](#), [summarizeSGP](#)

Examples

```
## Not run:
outputData <- outputSGP(Demonstration_Data)

## End(Not run)
```

```
prepareSGP
```

Prepare data for SGP analyses

Description

Utility function/exemplar used to embed supplied long data into a list object as a keyed data.table. NOTE: This function also serves the purposes of running many checks on the SGP object you construct to make sure it is up to date and in the best shape possible. If you have an older object that you wish to make sure is up to date with the latest version of the SGP package, running [prepareSGP](#) on an object is never bad thing to do. See examples for more information.

Usage

```
prepareSGP(data,
  data_supplementary=NULL,
  state=NULL,
  var.names=NULL,
  create.additional.variables=TRUE,
  fix.duplicates="keep.all")
```

Arguments

data	A panel data set in long form or an object of class SGP. See embedded sgpData_LONG data set for an exemplar.
data_supplementary	Supplementary data (e.g., student teacher lookup tables) to be embedded in SGP object in slot @Data_Supplementary. Data must be embedded in a list. Default is no data supplied.
state	A two letter acronymn indicating the state associated with the data. If not supplied, the function will try to infer what the state is from the data object name supplied.

<code>var.names</code>	A list or a dataframe that includes all required columns that do not match the SGP conventions, as well as all secondary columns needed for summarization and reporting.
<code>create.additional.variables</code>	Boolean argument indicating whether <code>prepareSGP</code> should create additional variables often used in analyses. For example, the function can create a variable <code>HIGH_NEED_STATUS</code> identifying the top and bottom quartile of students in each school by year by content area by grade grouping.
<code>fix.duplicates</code>	Argument for future functionality intended to deal with duplicate cases in supplied long data.

Value

Function returns an object of class `SGP`. The long data is in the data slot.

Author(s)

Ben Domingue <ben.domingue@gmail.com> and Damian W. Betebenner <dbetebenner@nciea.org>

See Also

[sgpData_LONG](#)

Examples

```
## Not run:
## prepareSGP is Step 1 of 5 of abcSGP
Demonstration_SGP <- prepareSGP(sgpData_LONG)

## Running prepareSGP on an already create SGP object as part of a annual update

Demonstration_SGP <- prepareSGP(Demonstration_SGP)

## End(Not run)
```

SGP-class

Class "SGP"

Description

The formal S4 class for SGP. This class stores the data object for use with the functions [analyzeSGP](#), [combineSGP](#), [summarizeSGP](#), and [visualizeSGP](#). The `SGP` class contains and organizes all the results associated with SGP analyses. [is.SGP](#) tests for membership for this class.

Details

list.null: combines class `list` and class `NULL`

SGP: The slot Names should be a data.frame with three columns. The first column (named "column") identifies the column in the original data for the associated variable. The second column is a character vector (named "nms.original") of original variable names for all of the variables in data. The third column is a character vector (named "nms.sgp") that identifies the capitalized variable names used with various functions from the SGP package. Original variables that are not needed for SGP functions should be marked as NA in the "nms.sgp" column.

Usage

Objects can be created by calls of the form `new("SGP", ...)`, but this is not encouraged. To instantiate a new instance of SGP class use the function [prepareSGP](#) instead.

`is.SGP(x)`

Slots

Data: A data.table including student-level data in a (long) format. Each VALID_CASE, CONTENT_AREA, YEAR, ID combination represents a unique case in the data. See `sgpData_LONG` for an exemplar data set

Data_Supplementary: A list (possibly NULL) providing additional data.tables containing student level multiple-membership lookup tables. For example, `sgpData_INSTRUCTOR_NUMBER` provides student teacher linkages and can be embedded in this slot using a list that contains it.

Names: A data.frame with three columns. The data.frame is used as a lookup table to translate state specific variable names to SGP variable names

SGP: A list including the output from [analyzeSGP](#)

Summary: A list including the output from [summarizeSGP](#)

Version: A list including the version of the SGP package used to construct the SGP object and the date the object was created.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com> and Damian W. Betebenner <dbetebenner@nciea.org>

See Also

[prepareSGP](#)

SGPstateData	<i>State assessment program data from large scale state assessments for use with SGP package</i>
--------------	--

Description

An environment (an object of class environment) containing information on state assessment programs, organized by state. Currently the environment contains achievement level cutscores and labels for the state assessments, assessment name and abbreviation, growth cutscores and labels, information on vertical scaling, and knots and boundaries currently being used for SGP analyses at the state level. The cutscores, in particular, are used to calculate growth-to-standard/projection values. States currently included in the data set are Arizona (AZ), California (CA), Colorado (CO), Connecticut (CT), Georgia (GA), Hawaii (HI), Idaho (ID), Indiana (IN), Kansas (KS), Maine (ME), Massachusetts (MA), Minnesota (MN), Mississippi (MS), Missouri (MO), Nevada (NV), New Hampshire (NH), New Jersey (NJ), New York (NY), Oregon (OR), Rhode Island (RI), Utah (UT), Vermont (VT), Virginia (VA), West Virginia (WV), Wisconsin (WI), Archdioces of Baltimore (AOB), Colorado English Language Assessment (CELA), Demonstration (DEMO), Australia NAPLAN (NAPLAN) and New Haven (NEW_HAVEN).

Usage

```
data(SGPstateData)
```

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <vaniwaarden@colorado.edu>

Source

State assessment data and technical assessment documentation

splineMatrix-class	<i>Class "splineMatrix"</i>
--------------------	-----------------------------

Description

The formal S4 class for coefficient matrices produced from the [studentGrowthPercentiles](#) function. This class stores the B-spline knots and boundaries used the coefficient matrix object for the production of student growth percentiles and projections.

Details

splineMatrix: This class contains the S3 matrix class, inheriting its methods. The slot `Knots` should be one or more lists of numeric vector(s) used in the internal call to `bs`, which generates a B-spline basis matrix from student scores. There are typically with 4 values for the knots. Similarly, `Boundaries` are used in `bs` for the `Boundary.knots` argument. This is always two values which are at or slightly beyond the lowest and highest observed student scores. `Content_Areas` and `Grade_Progression` provide information about the data (sub)set used to produce the matrix.

Objects from the Class

Objects can be created by calls of the form `new("splineMatrix", ...)`, but this is not encouraged. Previously produced coefficient matrices **MUST** be bound to the IDENTICAL knots and boundaries used to create them. Use the function [studentGrowthPercentiles](#) instead.

Slots

.Data: A coefficient matrix derived from [studentGrowthPercentiles](#).

Knots: A list(s) of numeric values used as the knots to generate the B-spline basis matrix in [studentGrowthPercentiles](#).

Boundaries: A list(s) of numeric values as the `Boundary.knots` to generate the B-spline basis matrix in [studentGrowthPercentiles](#).

Content_Areas: A list of content area names included in the data used to produce the coefficient matrix.

Grade_Progression: A list of the grades included in the data used to produce matrices.

Time: A list of the Times (e.g., years) measurements occurred included in the data used to produce matrices.

Time_Lags: A list of the time lags/differences between Time (e.g., years) included in the data used to produce matrices.

Version: A list including the version of the SGP package used to construct the `splineMatrix` object and the date the object was created.

Author(s)

Adam Van Iwaarden <Vaniwaarden@colorado.edu>, Ben Domingue <ben.domingue@gmail.com> and Damian W. Betebenner <dbetebenner@nciea.org>

See Also

[studentGrowthPercentiles](#)


```

        print.time.taken=TRUE,
        parallel.config=NULL,
calculate.simex=NULL,
verbose.output=FALSE)

```

Arguments

- panel.data** REQUIRED. Object of class list, data.frame, or matrix containing longitudinal student data in wide format. If supplied as part of a list, data should be contained in `panel.data$Panel_Data`. Data must be formatted so that student ID is the first variable/column, student grade/time variables for each time period, from earliest to most recent, are the next variables/columns, and student scale score variables for each year, from earliest to latest, are the remaining variables/columns. See [sgpData](#) for an exemplar data set. NOTE: The column position of the variables IS IMPORTANT, NOT the names of the variables.
- sgp.labels** REQUIRED. A list, `sgp.labels`, of the form `list(my.year= , my.subject=)` or `list(my.year= , my.subject= , my.extra.label)`. The user-specified values are used to save the student growth percentiles, coefficient matrices, knots/boundaries, and goodness of fit results in an orderly fashion using an appropriate combination of year & subject & grade. Except in special circumstances, supplying `my.year` and `my.subject` are sufficient to uniquely label derivative output.
- panel.data.vnames** Vector of variables to use in student growth percentile calculations. If not specified, function attempts to use all available variables.
- additional.vnames.to.return** A list of the form `list(VARIABLE_NAME_SUPPLIED=VARIABLE_NAME_TO_BE_RETURNED)` indicating data to be returned with results from [studentGrowthPercentiles](#) analyses.
- grade.progression** Preferred argument to specify a student grade/time progression in the data. For example, `3:4` would indicate to subset the data where the two most recent grades for which data are available are 3 and 4, respectively. The argument allows for non-sequential grade progressions to be analyzed with automatic removal of columns where "holes" occur in the supplied `grade.progression`. For example, for the `grade.progression c(7,8,10)`, the penultimate `GRADE` and `SCALE_SCORE` column in the supplied `panel.data` would be removed. The argument can also be combined with an appropriate `panel.data.vnames` argument to remove a year of data would analyze students progressing from 7 to 8 to 10.
- content.area.progression** Character vector of content area names of same length as `grade.progression` to be provided if not all identical to 'my.subject' in `sgp.labels` list. Vector will be used to populate the `@Content_Areas` slot of the `splineMatrix` class coefficient matrices. If missing, 'sgp.labels\$my.subject' is repeated in a vector length equal to `grade.progression`.
- year.progression** Character vector of years associated with grade and content area progressions.

If missing then the year.progression is assumed to end in 'my.year' provided in sgp.labels and be of the same length as grade.progression. Vector will be used to populate the @Years slot of the splineMatrix class coefficient matrices.

year.progression.lags	A numeric vector indicating the time lags/span between observations in the columns supplied to 'studentGrowthPercentiles'. The default, NULL, allows the function to calculate the lags/differences based upon the supplied years.
num.prior	Number of prior scores one wishes to use in the analysis. Defaults to num.panels-1. If num.prior=1, then only 1st order growth percentiles are computed, if num.prior=2, then 1st and 2nd order are computed, if num.prior=3, 1st, 2nd, and 3rd ... NOTE: specifying num.prior is necessary in some situations (in early grades for example) where the number of prior data points is small compared to the number of panels of data.
max.order.for.percentile	A positive integer indicating the maximum order for percentiles desired. Similar limiting of number of priors used can be accomplished using the grade.progression argument.
subset.grade	Student grade level for sub-setting. If the data fed into the function contains multiple grades, setting subset.grade=5 selects out those students in grade five in the most recent year of the data. If no sub-setting is desired, argument do not include the subset.grade argument. If grade.progression is supplied, then a subset grade is implicitly specified.
percentile.cuts	Additional percentile cuts (supplied as a vector) between 1 and 99 associated with each student's conditional distribution. Default is to provide NO growth percentile cuts (scale scores associated with those growth percentiles) for each student.
growth.levels	A two letter state acronym or a list of the form list(my.cuts= , my.levels=) specifying a vector of cuts between 1 and 99 (e.g., 35, 65) and the associated qualitative levels associated with the cuts (e.g., low, typical, and high). Note that the length of my.levels should be one more than the length of my.cuts. To add your growth levels to the SGPstateData data set, please contact the package administrator.
use.my.knots.boundaries	A list of the form list(my.year= , my.subject=) specifying a set of pre-calculated knots and boundaries for B-spline calculations. Most often used to utilize knots and boundaries calculated from a previous analysis. Knots and boundaries are stored (and must be made available) with panel.data supplied as a list in panel.data\$Knots_Boundaries\$my.subject.my.year. As of SGP_0.0-6 user can also supply a two letter state acronym to utilize knots and boundaries within the SGPstateData data set supplied with the SGP package. To add your knots and boundaries to the SGPstateData data set, please contact the package administrator. If missing, function automatically calculates knots, boundaries, and loss.hoss values and stores them in panel.data\$Knots_Boundaries\$my.subject.my.year where my.subject and my.year are provided by sgp.labels.
use.my.coefficient.matrices	A list of the form list(my.year= , my.subject=) specifying a set of pre-calculated coefficient matrices to use for student growth percentile calculations.

Can be used to calculate baseline referenced student growth percentiles or to calculate student growth percentiles for small groups of excluded students without recalculating an entire set of data. If missing, coefficient matrices are calculated based upon the provided data and stores them in `panel.data$Coefficient_Matrices$my.subject.my.year` where `my.subject` and `my.year` are provided by `sgp.labels`.

`calculate.confidence.intervals` A character vector providing either a state acronym or a variable name from the supplied panel data. If a state acronym, CSEM tables from the embedded [SGPstateData](#) (note: CSEM data must be embedded in the `SGPstateData` set. To have your state CSEMs embed in the `SGPstateData` set, please contact the package administrator) will be used. If a variable name, the supplied panel data must contain a variable providing student level CSEMs (e.g., with adaptive testing). NOTE: If a variable name is supplied, the user must also use the argument `panel.data.vnames` indicating what variables in the supplied `panel.data` will be used for the `studentGrowthPercentile` analysis. For greater control, the user can also supply a list of the form `list(state= , confidence.quantiles= , simulation.iterations=)` or `list(variable= , confidence.quantiles= , simulation.iterations= , distribution= , specifying the state or variable to use, confidence.quantiles to report from the simulated SGPs calculated for each student, simulation.iterations indicating the number of simulated SGPs to calculate, distribution indicating whether to the the Normal or Skew-Normal to calculate SGPs, and round (defaults to 1, which is an integer - see round_any from plyr package for details) giving the level to round to. If requested, simulations are calculated and simulated SGPs are stored in panel.data$Simulated_SGPs.`

`print.other.gp` Boolean argument (defaults to FALSE) indicating whether growth percentiles of all orders should be returned. The default returns only the highest order growth percentile for each student.

`print.sgp.order` Boolean argument (defaults to FALSE) indicating whether the order of the growth percentile should be provided in addition to the SGP itself.

`calculate.sgps` Boolean argument (defaults to TRUE) indicating whether student growth percentiles should be calculated following coefficient matrix calculation.

`rq.method` Argument to define the estimation method used in the quantile regression calculations. The default is the the "br" method referring to the Barrodale and Robert's L1 estimation detailed in Koenker (2005) and in the help for the quantile regression package.

`knot.cut.percentiles` Argument that specifies the quantiles to be used for calculation of B-spline knots. Default is to place knots at the 0.2, 0.4, 0.6, and 0.8 quantiles.

`knots.boundaries.by.panel` Boolean argument (defaults to FALSE) indicating whether knots and boundaries should be calculated by panel in supplied panel data instead of aggregating across panel. If panels are on different scales, then different knots and boundaries may be required to accommodate quantile regression analyses.

`exact.grade.progression.sequence` Boolean argument indicating whether the `grade.progression` supplied is used exactly (TRUE) as supplied or whether lower order analyses are run as part of the

	whole analysis (FALSE—the default).
<code>drop.nonsequential.grade.progression.variables</code>	Boolean argument indicating whether to drop variables that do not occur with a non-sequential grade progress. For example, if the grade progression 7, 8, 10 is provided, the penultimate variable in <code>panel.data</code> is dropped. Default is TRUE.
<code>convert.0and100</code>	Boolean argument (defaults to TRUE) indicating whether conversion of growth percentiles of 0 and 100 to growth percentiles of 1 and 99, respectively, occurs. The default produces growth percentiles ranging from 1 to 99.
<code>sgp.quantiles</code>	Argument to specify quantiles for quantile regression estimation. Default is Percentiles. User can additionally submit a vector of quantiles (between 0 and 1). Goodness of fit output only available currently for PERCENTILES.
<code>sgp.loss.hoss.adjustment</code>	Argument to control whether SGP is calculated using <code>which.max</code> for values associated with the <code>hoss</code> embedded in <code>SGPstateData</code> . Providing two letter state acronym utilizes this adjustment whereas supply NULL (the default) uses no adjustment.
<code>percuts.digits</code>	Argument specifying how many digits (defaults to 2) to print percentile cuts (if asked for) with.
<code>isotonize</code>	Boolean argument (defaults to TRUE) indicating whether quantile regression results are isotonized to prevent quantile crossing following the methods derived by Dette & Volgushev (2008).
<code>convert.using.loss.hoss</code>	Boolean argument (defaults to TRUE) indicating whether requested percentile cuts are adjusted using the lowest obtainable scale score (LOSS) and highest obtainable scale score (HOSS). Those percentile cuts above the HOSS are replaced with the HOSS and those percentile cuts below the LOSS are replaced with the LOSS. The LOSS and HOSS are obtained from the <code>loss</code> and <code>hoss</code> calculated with the knots and boundaries used for spline calculations.
<code>goodness.of.fit</code>	Boolean argument (defaults to TRUE) indicating whether to produce goodness of fit results associated with produced student growth percentiles. Goodness of fit results are <code>grid.grobs</code> stored in <code>panel.data\$Goodness_of_Fit</code> <code>\$my.subject.my.year</code> where <code>my.subject</code> and <code>my.year</code> are provided by <code>sgp.labels</code> .
<code>return.prior.scale.score</code>	Boolean argument (default to TRUE) indicating whether to include the prior scale score in the SGP data output. Useful for examining relationship between prior achievement and student growth.
<code>return.prior.scale.score.standardized</code>	Boolean argument (default to TRUE) indicating whether to include the standardized prior scale score in the SGP data output. Useful for examining relationship between prior achievement and student growth.
<code>return.norm.group.identifier</code>	Boolean argument (default to TRUE) indicating whether to include the content areas and years that form students' specific norm group in the SGP data output.

`print.time.taken` Boolean argument (defaults to TRUE) indicating whether to print message indicating information on studentGrowthPercentile analysis and time taken.

`parallel.config` parallel configuration argument allowing for parallel analysis by 'tau'. Defaults to NULL.

`calculate.simex` A character state acronymn or list including state/csem variable, simulation.iterations, lambda and extrapolation method. Defaults to NULL, no simex calculations performed.

`verbose.output` A boolean argument indicating whether the function should output verbose diagnostic messages.

Details

Typical use of the function is to submit a data frame to the function containing records of all students across all grades, allowing the function to subset out specific grade progressions using `grade.progression`. Additional uses include using pre-calculated results to recalculate SGPs for baseline referencing. [studentGrowthPercentiles](#) examples provide code for use in analyzing assessment data across multiple grades.

Value

Function returns an object of class list containing objects: `Coefficient_Matrices`, `Goodness_of_Fit`, `Knots_Boundaries`, `Panel_Data`, `SGPercentiles`, `Simulated_SGPs`.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <vaniwaarden@colorado.edu>

References

- Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.
- Betebenner, D. W. (2009). Norm- and criterion-referenced student growth. *Educational Measurement: Issues and Practice*, 28(4):42-51.
- Betebenner, D. W. (2008). Toward a normative understanding of student growth. In K. E. Ryan & L. A. Shepard (Eds.), *The Future of Test Based Accountability* (pp. 155-170). New York: Routledge.
- Dette, H. & Volgushev, S. (2008). Non-crossing non-parametric estimates of quantile curves. *Journal of the Royal Statistical Society B*, 70(3), 609-627.
- Koenker, R. (2005). *Quantile regression*. Cambridge: Cambridge University Press.

See Also

[studentGrowthProjections](#), [sgpData](#), [sgpData_LONG](#), [SGPstateData](#)

Examples

```
## Not run:
## Calculate 4th grade student growth percentiles using included sgpData

sgp_g4 <- studentGrowthPercentiles(panel.data=sgpData,
                                   sgp.labels=list(my.year=2013, my.subject="Reading"),
                                   percentile.cuts=c(1,35,65,99),
                                   subset.grade=4,
                                   num.prior=1)

## NOTE: "grade.progression" can be used in place of "subset.grade" and "num.prior"

sgp_g4_v2 <- studentGrowthPercentiles(panel.data=sgpData,
                                       sgp.labels=list(my.year=2013, my.subject="Reading"),
                                       percentile.cuts=c(1,35,65,99),
                                       grade.progression=c(3,4))

identical(sgp_g4$SGPercentiles, sgp_g4_v2$SGPercentiles)

## Established state Knots and Boundaries are available in the supplied SGPstateData
## file and used by supplying the appropriate two letter state acronym.

sgp_g4_DEMO <- studentGrowthPercentiles(panel.data=sgpData,
                                         sgp.labels=list(my.year=2013, my.subject="Reading"),
                                         use.my.knots.boundaries="DEMO",
                                         grade.progression=c(3,4))

## Sample code for running non-sequential grade progression analysis.

sgp_g8_DEMO <- studentGrowthPercentiles(panel.data=sgpData,
                                         sgp.labels=list(my.year=2013, my.subject="Reading"),
                                         use.my.knots.boundaries="DEMO",
                                         grade.progression=c(5,6,8))

## NOTE: Goodness of Fit results are stored as graphical objects in the
## Goodness_of_Fit slot. To view or save (using any R output device) try:
## Load 'grid' package to access grid.draw function

require(grid)
grid.draw(sgp_g4$Goodness_of_Fit$READING.2013$GRADE_4)

require(grid)
pdf(file="Grade_4_Reading_2013_GOF.pdf", width=8.5, height=4.5)
  grid.draw(sgp_g4$Goodness_of_Fit$READING.2013$GRADE_4)
dev.off()

# Other grades

sgp_g5 <- studentGrowthPercentiles(panel.data=sgpData,
                                   sgp.labels=list(my.year=2013, my.subject="Reading"),
```

```

        percentile.cuts=c(1,35,65,99),
        grade.progression=3:5)

sgp_g6 <- studentGrowthPercentiles(panel.data=sgpData,
    sgp.labels=list(my.year=2013, my.subject="Reading"),
    percentile.cuts=c(1,35,65,99),
    grade.progression=3:6)

sgp_g7 <- studentGrowthPercentiles(panel.data=sgpData,
    sgp.labels=list(my.year=2013, my.subject="Reading"),
    percentile.cuts=c(1,35,65,99),
    grade.progression=3:7)

sgp_g8 <- studentGrowthPercentiles(panel.data=sgpData,
    sgp.labels=list(my.year=2013, my.subject="Reading"),
    percentile.cuts=c(1,35,65,99),
    grade.progression=4:8)

## All output of studentGrowthPercentiles (e.g., coefficient matrices) is contained
## in the object. See, for example, names(sgp_g8), for all included objects.
## Results are stored in the slot SGPercentiles.

# Combine all results

sgp_all <- rbind(sgp_g4$SGPercentiles$READING.2013,
    sgp_g5$SGPercentiles$READING.2013,
    sgp_g6$SGPercentiles$READING.2013,
    sgp_g7$SGPercentiles$READING.2013,
    sgp_g8$SGPercentiles$READING.2013)

# Save SGP results to .csv file

write.csv(sgp_all, file="sgp_all.csv", row.names=FALSE, quote=FALSE, na="")

## NOTE: studentGrowthPercentiles ADDs results to the current SGP object.
## This allows one to "recycle" the object for multiple grades and subjects as desired.

# Loop to calculate all SGPs for all grades without percentile cuts
# but with growth levels

my.grade.sequences <- list(3:4, 3:5, 3:6, 3:7, 4:8)
my.sgpData <- list(Panel_Data=sgpData) ### Put sgpData into Panel_Data slot

for (i in seq_along(my.grade.sequences)) {
    my.sgpData <- studentGrowthPercentiles(panel.data=my.sgpData,
        sgp.labels=list(my.year=2013, my.subject="Reading"),
        growth.levels="DEMO",
        goodness.of.fit="DEMO",
        grade.progression=my.grade.sequences[[i]])
}

# Save Student Growth Percentiles results to a .csv file:

```

```

write.csv(my.sgpData$SGPercentiles$READING.2013,
file="2013_Reading_SGPercentiles.csv", row.names=FALSE, quote=FALSE, na="")

# Create PDFs of all Goodness of Fit results:

for (i in names(my.sgpData$Goodness_of_Fit$READING.2013)) {
  pdf(file=paste(i, "_Reading_2013_GOF.pdf", sep=""), width=8.5, height=11)
  grid.draw(my.sgpData[["Goodness_of_Fit"]][["READING.2013"]][i])
  dev.off()
}

## Loop to calculate all SGPs for all grades using 2006 to 2009 data

my.grade.sequences <- list(3:4, 3:5, 3:6, 3:7, 4:8)

for (i in seq_along(my.grade.sequences)) {
  my.sgpData_2009 <- studentGrowthPercentiles(panel.data=my.sgpData,
panel.data.vnames=c("ID", "GRADE_2007",
"GRADE_2008", "GRADE_2009", "GRADE_2010",
"SS_2007", "SS_2008", "SS_2009", "SS_2010"),
sgp.labels=list(my.year=2010, my.subject="Reading"),
grade.progression=my.grade.sequences[[i]])
}

## Loop to calculate all SGPs for all grades WITH 80

my.grade.sequences <- list(3:4, 3:5, 3:6, 3:7, 4:8)

for (i in seq_along(my.grade.sequences)) {
  my.sgpData <- studentGrowthPercentiles(panel.data=my.sgpData,
sgp.labels=list(my.year=2013, my.subject="Reading"),
calculate.confidence.intervals=list(state="DEMO",
confidence.quantiles=c(0.1, 0.9), simulation.iterations=100,
distribution="Normal", round=1),
grade.progression=my.grade.sequences[[i]])
}

### Example showing how to use pre-calculated coefficient
### matrices to calculate student growth percentiles

my.grade.sequences <- list(3:4, 3:5, 3:6, 3:7, 4:8)
my.sgpData <- list(Panel_Data=sgpData) ### Put sgpData into Panel_Data slot

for (i in seq_along(my.grade.sequences)) {
  my.sgpData <- studentGrowthPercentiles(panel.data=my.sgpData,
sgp.labels=list(my.year=2013, my.subject="Reading"),
growth.levels="DEMO",
grade.progression=my.grade.sequences[[i]])
}

```

```

percentiles.1st.run <- my.sgpData$SGPercentiles$READING.2013

### my.sgpData has as full set of coefficient matrices for Reading, 2013. To view these
names(my.sgpData$Coefficient_Matrices$READING.2013)

## Let's NULL out the SGPercentiles slot and recreate the percentiles
## using the embedded coefficient matrices

my.sgpData$SGPercentiles$READING.2013 <- NULL

for (i in seq_along(my.grade.sequences)) {
  my.sgpData <- studentGrowthPercentiles(panel.data=my.sgpData,
    sgp.labels=list(my.year=2013, my.subject="Reading"),
    use.my.knots.boundaries=list(my.year=2013, my.subject="Reading"),
    use.my.coefficient.matrices=list(my.year=2013, my.subject="Reading"),
    growth.levels="DEMO",
    grade.progression=my.grade.sequences[[i]])
}

percentiles.2nd.run <- my.sgpData$SGPercentiles$READING.2013

identical(percentiles.1st.run, percentiles.2nd.run)

## End(Not run)

```

studentGrowthPlot

Create a student growth and achievement chart

Description

Function used to produce individual student growth and achievement time lines/plots for based upon output from student growth percentile analyses. Function is integrated with SGPstateData to accommodate state specific scales and nomenclature including performance level names. See Betebenner (2012) for discussion

Usage

```

studentGrowthPlot(Scale_Scores,
  Plotting_Scale_Scores,
  Achievement_Levels,
  SGP,
  SGP_Levels,
  Grades,
  Cuts_NY1,
  Connect_Points="Arrows",
  Cutscores,
  Report_Parameters)

```

Arguments

Scale_Scores	A vector of historical scale scores.
Plotting_Scale_Scores	A vector of scale scores used as the vertical coordinates for plotting. If supplied, Scale_Scores are used for text and Plotting_Scale_Scores are used for the actual coordinates.
Achievement_Levels	A vector of historical performance levels.
SGP	A vector of historical student growth percentiles.
SGP_Levels	A vector of historical growth (SGP) levels (e.g., low, typical, high).
Grades	A vector of historical grades student was tested in.
Cuts_NY1	A vector of cuts scores for next year.
Connect_Points	Argument specifying how historical achievement scores are connected in the studentGrowthPlot. Currently accepts either Arrows, the default, or None.
Cutscores	A data.frame of long formatted achievement level cutscores.
Report_Parameters	A list containing arguments: Current_Year, Content_Area, State.

Details

Function currently used as part of SGP package to produce student growth charts for states. More documentation to follow.

Value

Returns a studentGrowthPlot graphical object for the content area specified. More documentation to follow.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

References

Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.

`studentGrowthPlot_Styles`*studentGrowth_Styles providing base templates for the core student-GrowthPlot function*

Description

Function includes five "styles" associated with `studentGrowthPlot` to create PDF student growth plots depicting the growth and achievement for an individual student over time. The five styles display one to five content area student growth plots on a single page. The one, two, and three content area plots are rendered on 8.5 by 11 PDFs and the four and five content area plots are rendered on 11 by 17 PDFs. These charts are currently being used in multiple states to report student growth results. This function is called by `visualizeSGP` to generate individual student reports. This function may expand in the future to incorporate other possible individual student reports.

Usage

```
studentGrowthPlot_Styles(  
  sgPlot.data,  
  state,  
  last.year,  
  content_areas,  
  districts,  
  schools,  
  reports.by.student,  
  reports.by.instructor,  
  reports.by.school,  
  sgPlot.years,  
  sgPlot.demo.report,  
  sgPlot.folder,  
  sgPlot.folder.names,  
  sgPlot.anonymize,  
  sgPlot.front.page,  
  sgPlot.header.footer.color,  
  sgPlot.fan,  
  sgPlot.cleanup,  
  sgPlot.baseline,  
  sgPlot.zip,  
  sgPlot.output.format)
```

Arguments

<code>sgPlot.data</code>	Wide formatted individual student report data used to produce student growth plots. To view structure of wide formatted data, utilize the <code>sgPlot.save.sgPlot.data</code> option with <code>visualizeSGP</code> to save wide formatted student growth plot data.
--------------------------	---

<code>state</code>	Acronym indicating state associated with the summaries for access to assessment program information embedded in <code>SGPstateData</code> .
<code>last.year</code>	Argument indicating the final year represented in the student growth plots.
<code>content_areas</code>	Argument providing the content areas depicted in the student growth plots.
<code>districts</code>	A vector of district numbers indicating which districts student growth plots should be produced for.
<code>schools</code>	A vector of school numbers indicating which schools student growth plots should be produce for.
<code>reports.by.student</code>	A boolean variable passed to <code>studentGrowthPlot_Styles</code> indicating whether separate individual plots will be produced or separate reports and a summary catalog containing those reports will be produced.
<code>reports.by.instructor</code>	A boolean variable passed to <code>studentGrowthPlot_Styles</code> indicating whether individual plots will be collated and bundled as a summary catalog by instructor.
<code>reports.by.school</code>	A boolean variable passed to <code>studentGrowthPlot_Styles</code> indicating whether individual plots will be collated and bundled as a summary catalog by school. Prior to version 0.9-9.7, this was the only way of bundling reports and was thus the default.
<code>sgPlot.years</code>	A vector of all years over which student growth plots are being produced.
<code>sgPlot.demo.report</code>	A boolean argument indicating whether a demonstration report catalog (with anonymized individual, school, and district names) is to be produced.
<code>sgPlot.folder</code>	A character argument specifying the folder into which the student growth reports will be placed.
<code>sgPlot.folder.names</code>	Either names or number indicating whether names or numbers should be used as folder names.
<code>sgPlot.anonymize</code>	A boolean argument indicating whether individual, school, and district names should be anonymized.
<code>sgPlot.front.page</code>	A character vector indicating the file, the the base directory, that should be used as the front page for the student growth plots.
<code>sgPlot.header.footer.color</code>	A color (as a character) specifying the header/footer color of the report.
<code>sgPlot.fan</code>	A boolean argument indicating whether the projection fan indicating growth rates necessary to reach 1 years targets be displayed.
<code>sgPlot.cleanup</code>	A boolean argument indicating whether to cleanup/remove files produced as part of pdfLaTeX build.
<code>sgPlot.baseline</code>	A boolean argument indicating whether to use baseline referenced student growth percentiles in student growth plots.

sgPlot.zip A boolean argument indicating whether to school folders.
 sgPlot.output.format Argument indicating the desired type of output format for student growth plots.
 Either 'PDF', 'PNG', or 'PDF_PIECES'.

Details

studentGrowthPlot_Styles is a utility function containing five templates for displaying [studentGrowthPlots](#). The templates display two or three content areas depending upon the availability of test data for the state. This function will expand in the future to allow for other types of individual growth and achievement reports.

Value

Function produces *numerous* (potentially hundreds of thousands of) PDF student growth plots.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <vaniwaarden@colorado.edu>

References

Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.

studentGrowthProjections

Student Growth Projections

Description

Function to calculate percentile growth projections/trajectories using large scale assessment data and results derived from student growth percentile calculation. Function can produce percentile growth trajectories, as well as growth percentiles, sufficient for each student to reach a set of predefined scale score cut.

Usage

```
studentGrowthProjections(panel.data,
                          sgp.labels,
                          grade.progression,
                          content_area.progression=NULL,
                          grade.projection.sequence=NULL,
                          content_area.projection.sequence=NULL,
                          max.forward.progression.years,
                          max.forward.progression.grade,
                          max.order.for.progression,
```

```

use.my.knots.boundaries,
use.my.coefficient.matrices,
panel.data.vnames,
achievement.level.prior.vname=NULL,
performance.level.cutscores,
calculate.sgps=TRUE,
convert.0and100=TRUE,
projection.unit="YEAR",
percentile.trajectory.values=NULL,
isotonize=TRUE,
lag.increment=0,
projcuts.digits=NULL,
print.time.taken=TRUE)

```

Arguments

- panel.data** Object of class list containing longitudinal student data in wide format in panel.data\$Panel_Data. See [studentGrowthPercentiles](#) for data requirements. List object must also contain panel.data\$Knots_Boundaries and panel.data\$Coefficient_Matrices. See [sgpData](#) for an exemplar data set. NOTE: The column position of the variables IS IMPORTANT, NOT the names of the variables.
- sgp.labels** REQUIRED. A list, sgp.labels, of the form list(my.year= , my.subject=). The user-specified values are used to save the percentile growth projections/trajectories and identify coefficient matrices and knots & boundaries for calculation if use.my.coefficient.matrices or use.my.knots.boundaries is missing. Partly replaces previous argument proj.function.labels.
- grade.progression** REQUIRED. Argument to specify a student grade/time progression in the data to be used for percentile growth projection/trajectory calculation. This argument helps in replacing previous arguments num.panels, max.num.scores, and num.prior.scores.
- content_area.progression** Argument to specify a student content area progression in the data supplied for percentile growth projection/trajectory calculation.
- grade.projection.sequence** Argument to manually supply grade sequence over which projection is made. Defaults to NULL and is calculated from available data.
- content_area.projection.sequence** Argument to manually supply content area sequence over which projection is made. Defaults to NULL and is calculated from available data.
- max.forward.progression.years** The MAXIMUM number of years/grades/time periods to project forward conditional upon available coefficient matrices. If missing, function will project forward as far as allowed by available coefficient matrices.
- max.forward.progression.grade** The MAXIMUM grade to project forward based upon available coefficient matrices. If missing, function will project forward as far as allowed by available coefficient matrices.

`max.order.for.progression`

Argument to specify the maximum coefficient matrix order to be used for percentile growth projection/trajectory calculation. If missing, the function utilizes the highest matrix order available.

`use.my.knots.boundaries`

A list of the form `list(my.year= , my.subject=)` specifying the set of pre-calculated knots and boundaries for B-spline calculations. Knot and boundaries are stored (and must be made available) with `panel.data` supplied as a list in `panel.data$Knots_Boundaries$my.year.my.subject`. As of SGP_0.0-6.9 user can also supply a two letter state acronym to utilize knots and boundaries within the `SGPstateData` data set supplied with the SGP package. If missing, function tries to retrieve knots and boundaries from `panel.data$Knots_Boundaries$my.year.my.subject` where `my.year` and `my.subject` are provided by `sgp.labels`.

`use.my.coefficient.matrices`

A list of the form `list(my.year= , my.subject=)` specifying the set of pre-calculated coefficient matrices to use for percentile growth projection/trajectory calculations. Coefficient matrices are stored (and must be available) with `panel.data` supplied as a list in `panel.data$Coefficient_Matrices$my.year.my.subject`. If missing, function tries to retrieve coefficient matrices from `panel.data$Coefficient_Matrices$my.year.my.subject` where `my.year` and `my.subject` are provided by `sgp.labels`.

`panel.data.vnames`

Vector of variables to use in percentile growth projection/trajectory calculations. If not specified, function attempts to use all available variables.

`achievement.level.prior.vname`

Character vector indicating variable is supplied panel data corresponding to the prior achievement level to be added to the output. Used in the production of growth to standard analyses.

`performance.level.cutscores`

Argument for supplying performance level cutscores to be used for determining growth-to-standard percentile growth trajectory calculations. Argument accepts a two letter state acronym (e.g., "CO") that retrieves cutscores that are automatically embedded in a data set contained in the SGP package. Argument also accepts a subject specific list of the form:

```
performance.level.cutscores <- list(
  Reading=list(GRADE_3=c(cut1, cut2, cut3),
                GRADE_4=c(cut1, cut2, cut3),
                . . .
                GRADE_8=c(cut1, cut2, cut3)),
  Math=list(GRADE_3=c(cut1, cut2, cut3),
            . . .
            GRADE_7=c(cut1, cut2, cut3),
            GRADE_8=c(cut1, cut2, cut3)))
```

Note that the subject name must match that provided by `sgp.labels`. If cuts are not desired leave the cutscore unspecified, which is the default. If your state's

	cutscores are not included in the SGPstateData data set or are incorrect, please contact <dbetebenner@nciea.org> to have them added or corrected!
calculate.sgps	Boolean argument (defaults to TRUE) indicating whether to calculate student growth projections. Currently used to bypass calculations in analyzeSGP when scale changes occur.
convert.0and100	Boolean argument (defaults to TRUE) indicating whether conversion of growth percentiles of 0 and 100 to growth percentiles of 1 and 99, respectively, occurs. The default produces growth percentiles ranging from 1 to 99.
projection.unit	Argument specifying the units in which the projections/trajectories and cuts are reported. Either "GRADE", the default, or "YEAR".
percentile.trajectory.values	An integer argument with values ranging from 1 to 100 that returns a vector of percentile trajectory cuts (default is NULL, no percentile trajectory values returned). The returned values are the lower bound for the interval associated with the percentile(s) requested. For example, by specifying 'percentile.trajectory.values=1' the user would receive the conditional .005 quantile associated with each student's distribution.
isotonize	Boolean argument (defaults to TRUE) indicating whether quantile regression results are isotonized to prevent quantile crossing following the methods derived by Dette & Volgushev (2008).
lag.increment	A non-negative integer (defaults to 0) indicating the lag associated with the data supplied for projections. Only relevant if Cutscores or Knots and Boundaries are year dependent.
projcuts.digits	The number of digits (defaults to NULL/0) percentile trajectories (if requested) are formatted.
print.time.taken	Boolean argument (defaults to TRUE) indicating whether to print message indicating information on studentGrowthProjection analysis and time taken.

Value

Function returns the input `panel.data` list object with the additional percentile growth trajectories/percentiles stored in `panel.data$SGProjections$my.year.my.subject` consisting of student IDs and the associated percentile growth projections/trajectories and cuts. The data frame contains projections/trajectories for each performance level cut-point supplied and each percentile cut the user specifies.

Note

Use of this function assumes prior calculation of student growth percentiles, making the coefficient matrices available within the `panel.data$Coefficient_Matrices` list object. Additionally, if cutscores are desired they must be supplied explicitly by the user (as detailed above in `performance.level.cutscores`) or included in the SGPstateData data set. If your state's cutscores are not included or are incorrect, please contact <dbetebenner@nciea.org> to have cutscores added or corrected!

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <vaniwaarden@colorado.edu>

References

- Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.
- Betebenner, D. W. (2009). Norm- and criterion-referenced student growth. *Educational Measurement: Issues and Practice*, 28(4):42-51.
- Betebenner, D. W. (2008). Toward a normative understanding of student growth. In K. E. Ryan & L. A. Shepard (Eds.), *The Future of Test Based Accountability* (pp. 155-170). New York: Routledge.
- Dette, H. & Volgushev, S. (2008). Non-crossing non-parametric estimates of quantile curves. *Journal of the Royal Statistical Society B*, 70(3), 609-627.

See Also

[studentGrowthPercentiles](#), [sgpData](#)

Examples

```
## Not run:
## First calculate SGPs for 2013
my.grade.sequences <- list(3:4, 3:5, 3:6, 3:7, 4:8)
my.sgpData <- list(Panel_Data = sgpData)

for (i in seq_along(my.grade.sequences)) {
  my.sgpData <- studentGrowthPercentiles(panel.data=my.sgpData,
    sgp.labels=list(my.year=2013, my.subject="Reading"),
    use.my.knots.boundaries="DEMO",
    grade.progression=my.grade.sequences[[i]])
}

## Calculate Growth Projections

my.grade.progressions <- list(3, 3:4, 3:5, 3:6, 4:7)

for (i in seq_along(my.grade.progressions)) {
  my.sgpData <- studentGrowthProjections(panel.data=my.sgpData,
    sgp.labels=list(my.year=2013, my.subject="Reading"),
    projcuts.digits=0,
    projection.unit="GRADE",
    performance.level.cutcores="DEMO",
    percentile.trajectory.values=c(25, 50, 75),
    grade.progression=my.grade.progressions[[i]])
}

## Save the Student Growth Projections Results to a .csv file:
write.csv(my.sgpData$SGProjections$READING.2013,
  file= "2013_Reading_SGProjections.csv", row.names=FALSE, quote=FALSE)
```

```
## End(Not run)
```

summarizeSGP	<i>Summarize student scale scores, proficiency levels and student growth percentiles according to user specified summary group variables</i>
--------------	--

Description

Utility function used to produce summary tables using long formatted data that contain student growth percentiles. An exemplar is provided from the successive execution of [prepareSGP](#), [analyzeSGP](#) and [combineSGP](#).

Usage

```
summarizeSGP(sgp_object,
             state,
             years,
             content_areas,
             sgp.summaries=NULL,
             summary.groups=NULL,
             confidence.interval.groups=NULL,
             produce.all.summary.tables=FALSE,
             summarizeSGP.baseline=NULL,
             projection.years.for.target=3,
             save.old.summaries=FALSE,
             parallel.config=NULL)
```

Arguments

<code>sgp_object</code>	A list containing long formatted data in the Student slot. If summaries of student growth percentiles are requested, those quantities must first be produced (possibly by first using analyzeSGP) and subsequently combined with the Student data (possibly with combineSGP).
<code>state</code>	Acronym indicating state associated with the summaries for access to assessment program information embedded in <code>SGPstateData</code> .
<code>years</code>	A vector indicating year(s) in which to produce summary tables associated with student growth percentile and percentile growth trajectory/projection analyses. If missing the function will use the data to calculate years and produce summaries for the most recent three years.
<code>content_areas</code>	A vector indicating content area(s) in which to produce student growth percentiles and/or student growth projections/trajectories. If missing the function will use the data to infer the content area(s) available for analyses.
<code>sgp.summaries</code>	A list giving the summaries requested for each group analyzed based upon the <code>summary.group</code> argument. The default (produced internal to <code>summarizeSGP</code>) summaries include:

MEDIAN_SGP	The group level median student growth percentile.
MEDIAN_SGP_COUNT	The number of students used to compute the median.
PERCENT_AT_ABOVE_PROFICIENT	The percentage of students at or above proficient.
PERCENT_AT_ABOVE_PROFICIENT_COUNT	The number of students used to compute the percentage at/above proficient.
PERCENT_AT_ABOVE_PROFICIENT_PRIOR	The percentage of students at or above proficient in the prior year.
PERCENT_AT_ABOVE_PROFICIENT_PRIOR_COUNT	The number of students used to compute the percentage at/above proficient.

NOTE: The internal function `percent_in_category()` summary function requires a variable that **MUST** be a factor with proficiency categories as levels. The function utilizes the `SGPstateData` with the provided state name in an attempt to identify achievement levels and whether or not they are considered proficient.

`summary.groups` A list consisting of 8 elements indicating the types of groups across which all summaries are taken (Inclusion means that summaries will be calculated for levels of the associated variable). For state data, if the list is not explicitly provided, the function will attempt to determine levels based upon meta data supplied in the `@Names` slot of the provided SGP object. See [prepareSGP](#) for more information on supplied meta-data.

<code>institution:</code>	State, District and/or School.
<code>content area:</code>	Variable indicating content area (default is <code>CONTENT_AREA</code>) if content area summaries are of interest.
<code>time:</code>	Variable indicating time (default is <code>YEAR</code>) if time summaries are of interest. NOTE: Cross year summaries are not supported.
<code>institution_type:</code>	Variable(s) indicating the type of institution (default <code>EMH_LEVEL</code>) if summaries by institution type are of interest.
<code>institution_level:</code>	Variable(s) indicating levels within the institution (default <code>GRADE</code>) if summaries by institution level are of interest.
<code>demographic:</code>	Demographics variables if summaries by demographic subgroup are of interest.
<code>institution_inclusion:</code>	Variables indicating inclusion for institutional calculations.
<code>growth_only_summary:</code>	Variables indicating whether to calculate summaries only for those students with growth in addition to the other variables.

All group slots **MUST** be included in the list, although `NULL` can be provided if a grouping subset is not desired. All possible combinations of the group variables are produced.

`confidence.interval.groups`

A list consisting of information used to calculate group confidence intervals:

TYPE:	
VARIABLES:	
QUANTILES	
GROUP	
content	
time	
institution_type	
institution_level	
demographic	
institution_inclusion	
growth_only_summary	The growth only summary variables if confidence intervals by growth only summary group are desired.

For CSEM analysis this argument requires that simulated SGPs have been produced (see [analyzeSGP](#) for more information). List slots set to NULL will not produce confidence intervals. NOTE: This is currently an experimental functionality and is very memory intensive. Groups to be included should be identified selectively! The default 95% confidence intervals are provided in the selected summary tables as two additional columns named LOWER_MEDIAN_SGP_95_CONF_BOUND and UPPER_MEDIAN_SGP_95_CONF_BOUND.

`produce.all.summary.tables`

A boolean variable, defaults to FALSE, indicating whether the function should produce ALL possible summary table. By default, a set of approximately 70 tables are produced that are used in other parts of the packages (e.g., bubblePlots).

`summarizeSGP.baseline`

A boolean variable, defaults to FALSE, indicating whether the function should utilize baseline sgp for summary table production. By default, a set of approximately 100 tables are produced that are used in other parts of the packages (e.g., bubblePlots).

`projection.years.for.target`

An integer argument indicating SGP_TARGET variables to summarize based upon years projected forward. Default is 3 years which is what is generally used by most states.

`save.old.summaries`

A boolean argument, defaults to FALSE, indicating whether to save the @Summary slot (if not NULL) prior to calculating new summaries. By defaulting to FALSE, the function overwrites previous (e.g., last year's summaries) summaries.

`parallel.config`

A named list with, at a minimum, two elements indicating 1) the BACKEND package to be used for parallel computation and 2) the WORKERS list to specify the number of processors to be used in each major analysis. The BACKEND element can be set = to FOREACH or PARALLEL. Please consult the manuals and vignettes for information of these packages! The [analyzeSGP](#) help page contains more thorough explanation and examples of the parallel.config setup.

TYPE is a third element of the parallel.config list that provides necessary information when using FOREACH or PARALLEL packages as the backend. With BACKEND="FOREACH", the TYPE element specifies the flavor of 'foreach' backend. As of version 1.0-1.0, only "doParallel" is supported. TYPE=NA (default) produces summaries sequentially. If BACKEND = "PARALLEL", the parallel package will be used. This package combines deprecated parallel packages snow and multicore. Using the "snow" implementation of parallel the function will create a cluster object based on the TYPE element specified and the number of workers requested (see WORKERS list description below). The TYPE element indicates the users preferred cluster type (either "SOCK" for socket cluster or "MPI" for an OpenMPI cluster). If Windows is the operating system, this "snow" implementation must be use and the TYPE element must = "SOCK". Defaults are assigned based on operating system if TYPE is missing based on system OS. Unix/Mac OS defaults to the "multicore" to avoid worker node prescheduling and appears to be more efficient in these operating systems.

The WORKERS element is a list with SUMMARY specifying the number of processors (nodes) desired or available. For example, SUMMARY=2 may be used on a dual core machine to use both cores available. (NOTE: choice of the number of cores is a balance between the number of processors available and the amount of RAM a system has; each system will be different and may require some adjustment).

Default is FOREACH as the back end, TYPE=NA and WORKERS=1, which produces summary tables sequentially: 'list(BACKEND="FOREACH", TYPE=NA, WORKERS=list(SUMMARY=1))'

Example parallel use cases are provided below.

Details

Function makes use of the foreach package to parallel process summary tables of student data. The proper choice of parallel backend is dependent upon the user's operating system, software and system memory capacity. Please see the foreach documentation for details. By default, the function will process the summary tables sequentially.

Value

Function returns lists containing the summary tables as data.table objects in the Summary slot of the SGP data object. Each institution has a slot in the Summary list.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <vaniwaarden@colorado.edu>

See Also

[prepareSGP](#), [analyzeSGP](#), [combineSGP](#)

Examples

```
## Not run:
## summarizeSGP is Step 4 of 5 of abcSGP
Demonstration_SGP <- sgpData_LONG
Demonstration_SGP <- prepareSGP(Demonstration_SGP)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP)
Demonstration_SGP <- combineSGP(Demonstration_SGP)
Demonstration_SGP <- summarizeSGP(Demonstration_SGP)

### Example uses of the parallel.config argument

## Windows users must use the parallel package and R version >= 2.13:
# Note the number of workers is 8, and SOCK type cluster is used.
# This example is would be good for a single workstation with 8 cores.
. . .
parallel.config=list(
  BACKEND="PARALLEL", TYPE="SOCK",
  WORKERS=list(SUMMARY=2))
. . .
```

```

# doParallel package - only available with R 2.13 or newer
. . .
parallel.config=list(
  BACKEND="FOREACH", TYPE="doParallel",
  WORKERS=list(SUMMARY=6))
. . .

## parallel package - only available with R 2.13 or newer
# Note the number of workers is 50, and MPI is used,
# suggesting this example is for a HPC cluster usage.
. . .
parallel.config=list(
  BACKEND="PARALLEL", TYPE="MPI"),
  WORKERS=list(SUMMARY=50))
. . .

# NOTE: This list of parallel.config specifications is NOT exhaustive.
# See examples in analyzeSGP documentation for some others.

## End(Not run)

```

testSGP

Test SGP Package functions

Description

testSGP runs large scale tests of the SGP package to test for consistent performance across releases.

Usage

```
testSGP(TEST_NUMBER,
  save.results=FALSE,
  memory.profile=FALSE)
```

Arguments

TEST_NUMBER	An integer indicating the test to be run. Type 'testSGP()' to see list and description of available tests.
save.results	A boolean variable, defaulting to FALSE, indicating whether the results of the analysis is saved to the working directory.
memory.profile	A boolean variable indicating whether to use memory profiling via Rprof. Experimental. Defaults to FALSE.

Value

Returns output associated with functions being run.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <vaniwaarden@colorado.edu>

See Also

[abcSGP](#)

Examples

```
## Not run:
## testSGP(1) runs abcSGP for all years in sgpData_LONG
testSGP(1)

## testSGP(2) runs abcSGP for all years in sgpData_LONG except most recent year than adds
## in most recent year and runs current year analyses
testSGP(2)

## End(Not run)
```

updateSGP

Function to update SGP object with additional year's analyses

Description

updateSGP takes an object of class SGP and adds in additional data (usually an additional year's data) and runs analyses on that additional year's data including the results in the supplied SGP object.

Usage

```
updateSGP(
  what_sgp_object=NULL,
  with_sgp_data_LONG=NULL,
  state=NULL,
  years=NULL,
  content_areas=NULL,
  save.old.summaries=TRUE,
  save.intermediate.results=FALSE,
  ...)
```

Arguments

what_sgp_object

The SGP object to which the additional data will be added and analyzed. This object must be specified.

`with_sgp_data_LONG` The additional data in LONG for to be added to the supplied object of class SGP analyzed. The additional data must be in the same form as the data in the @Data slot. If `with_sgp_data_LONG` is not supplied, the function with `update` the `sgp_object` supplied in 'what_sgp_object' using the embedded coefficient matrices, essentially re-doing the analyses.

`state` The 'state' for the `sgp_object`. Derived from `sgp_object` name if not explicitly supplied.

`years` If only 'what_sgp_object' is supplied, `years` specifies the years to be run amongst those in the provided `sgp_object`.

`content_areas` If only 'what_sgp_object' is supplied, `content_areas` specifies the content areas to be run amongst those provided by the coefficient matrices in the `sgp_object`. Default is to run all analyses associated with the coefficient matrices.

`save.old.summaries` A boolean argument (defaults to FALSE which will delete the @Summary slot before creating new summaries) indicating whether the call to `summarizeSGP` should save existing summaries in the @Summary slot.

`save.intermediate.results` A boolean argument (defaults to FALSE) indicating whether results should be save to the current directory after each step of the analysis.

... Arguments to be passed to `abcSGP` for finer control over SGP calculations. For example, 'parallel.config' can be supplied to `abcSGP` for parallel computation.

Value

Returns and object of class SGP including additional analyses.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

See Also

`prepareSGP` and `abcSGP`

Examples

```
## Not run:
### Run analyses on all but final year's of data

Demonstration_Data_LONG <- subset(sgpData_LONG, YEAR
Demonstration_Data_LONG_2012_2013 <- subset(sgpData_LONG, YEAR

Demonstration_SGP <- abcSGP(
  sgp_object=Demonstration_Data_LONG,
  sgPlot.demo.report=TRUE)

### Run updateSGP on Demonstration_SGP and the 2012_2013 data
```

```
Demonstration_SGP <- updateSGP(
  what_sgp_object=Demonstration_SGP,
  with_sgp_data_LONG=Demonstration_Data_LONG_2012_2013)

## End(Not run)
```

visualizeSGP

Visualize data from SGP analyses

Description

Utility function to produce a variety of graphical displays associated with student growth percentile/percentile growth trajectory results. Function currently includes facility to produce individual student growth and achievement plots, interactive bubble plots depicting summary growth and achievement data, and growth and achievement charts showing system level growth and achievement data as shown on the cover of *Educational Measurement: Issues and Practice* as part of Betebenner (2009) and Betebenner (2012).

Usage

```
visualizeSGP(
  sgp_object,
  plot.types=c("bubblePlot", "studentGrowthPlot", "growthAchievementPlot"),
  state,
  bPlot.years=NULL,
  bPlot.content_areas=NULL,
  bPlot.districts=NULL,
  bPlot.schools=NULL,
  bPlot.instructors=NULL,
  bPlot.styles=c(1),
  bPlot.levels=NULL,
  bPlot.level.cuts=NULL,
  bPlot.full.academic.year=TRUE,
  bPlot.minimum.n=10,
  bPlot.anonymize=FALSE,
  bPlot.prior.achievement=TRUE,
  bPlot.draft=FALSE,
  bPlot.demo=FALSE,
  bPlot.format="print",
  bPlot.folder="Visualizations/bubblePlots",
  sgPlot.save.sgPlot.data=FALSE,
  sgPlot.years=NULL,
  sgPlot.content_areas=NULL,
  sgPlot.districts=NULL,
  sgPlot.schools=NULL,
  sgPlot.reports.by.school=TRUE,
```

```

sgPlot.instructors=NULL,
sgPlot.reports.by.instructor=FALSE,
sgPlot.students=NULL,
sgPlot.reports.by.student=FALSE,
sgPlot.header.footer.color="#4CB9CC",
sgPlot.front.page=NULL,
sgPlot.folder="Visualizations/studentGrowthPlots",
sgPlot.folder.names="number",
sgPlot.fan=TRUE,
sgPlot.anonymize=FALSE,
sgPlot.cleanup=TRUE,
sgPlot.demo.report=FALSE,
sgPlot.produce.plots=TRUE,
sgPlot.baseline=NULL,
sgPlot.zip=TRUE,
sgPlot.output.format="PDF",
sgPlot.show.targets.years.forward=NULL,
gaPlot.years=NULL,
gaPlot.content_areas=NULL,
gaPlot.students=NULL,
gaPlot.format="print",
gaPlot.baseline=NULL,
gaPlot.max.order.for.progression=NULL,
gaPlot.folder="Visualizations/growthAchievementPlots",
parallel.config=NULL)

```

Arguments

<code>sgp_object</code>	An object of class SGP containing long formatted data in the Data slot that will be used for the production of student growth and achievement plots and system growth and achievement plots, summary data from <code>summarizeSGP</code> in the Summary slot for bubble plots.
<code>plot.types</code>	A character vector indicating what types of plots to construct. Currently available plots include <code>bubblePlot</code> , <code>studentGrowthPlot</code> , and <code>growthAchievementPlot</code> .
<code>state</code>	Acronym indicating state associated with the summaries for access to assessment program information embedded in <code>SGPstateData</code> .
<code>bPlot.years</code>	A vector indicating year(s) in which to produce <code>bubblePlots</code> using data available in <code>sgp_object</code> . If missing the function will use the last year available in the data to produce <code>bubblePlots</code> .
<code>bPlot.content_areas</code>	A vector indicating content area(s) to produce <code>bubblePlots</code> using data available in <code>sgp_object</code> . If missing the function will produce plots for all available content areas provided in the data.
<code>bPlot.districts</code>	A vector indicating districts to produce <code>bubblePlots</code> using data available in <code>sgp_object</code> . Consult <code>bubblePlot</code> styles to determine which <code>bubblePlots</code> styles accept specification for districts. Default is to produce plots for all available districts in the data.

<code>bPlot.schools</code>	A vector indicating schools to produce <code>bubblePlots</code> using data available in <code>sgp_object</code> . Consult <code>bubblePlot</code> styles to determine which <code>bubblePlot</code> styles accept specification for schools. Default is to produce plots for all available schools in the data.
<code>bPlot.instructors</code>	A vector indicating instructors to produce <code>bubblePlots</code> using data available in <code>sgp_object</code> . If missing the function will produce plots for all available instructors provided in the data where schools and districts represent relevant units to be represented by the specific <code>bubblePlot</code> style.
<code>bPlot.styles</code>	A vector of positive integers indicating the types of <code>bubblePlots</code> to produce using data available in <code>sgp_object</code> . See associated documentation for example plots.
<code>bPlot.levels</code>	A character vector of levels to be used to color bubbles in the <code>bubblePlot</code> . See associated documentation for example plots.
<code>bPlot.level.cuts</code>	A vector of cuts to be used to distinguish levels used to color bubbles in the <code>bubblePlot</code> . See associated documentation for example plots.
<code>bPlot.full.academic.year</code>	A boolean argument (defaults to TRUE) indicating whether <code>bubblePlots</code> should use full academic year results if available.
<code>bPlot.minimum.n</code>	A positive integer (defaults to 10) indicating the minimum size for summary values to be displayed in the <code>bubblePlots</code> .
<code>bPlot.anonymize</code>	A boolean argument (defaults to FALSE) indicating whether to anonymize <code>bubblePlots</code> school and district names that appear in the plots and data tips of the plots. For student level anonymization, the function utilizes the <code>randomNames</code> package to produce gender and ethnic correct names based upon gender and ethnicity codes available in <code>sgp_object@Data</code> .
<code>bPlot.prior.achievement</code>	A boolean argument (defaults to TRUE) indicating whether to produce <code>bubblePlots</code> using prior achievement as well as current achievement as the vertical dimension of the <code>bubblePlot</code> .
<code>bPlot.draft</code>	A boolean argument (defaults to FALSE) indicating whether to put an indicator on the chart noting that the results are draft and to not distribute.
<code>bPlot.demo</code>	A boolean argument (defaults to FALSE) indicating whether to produce demo student level plots (styles 150 and/or 153) for instructors.
<code>bPlot.format</code>	Either "print" or "presentation" indicating whether to optimize the plot for print form (light background) or presentation form (dark background).
<code>bPlot.folder</code>	Character vector indicating where <code>bubblePlots</code> should be placed. Default folder is "Visualizations/bubblePlots".
<code>sgPlot.save.sgPlot.data</code>	A boolean argument indicating whether wide formatted data should be save as part of individual student report production. For operational student report production construction, using a wide formatted file for a large state can save in having to reshape the same long file multiple times.

- `sgPlot.years` A vector indicating year for which to produce `studentGrowthPlots`. The supplied year indicates the *final* year associated with each student's `studentGrowthPlot`. If missing the function will use the last year available in the data to produce `studentGrowthPlots`.
- `sgPlot.content_areas` A vector indicating the content areas for which to produce `studentGrowthPlots`. If missing, the function will utilize all available content areas.
- `sgPlot.districts` A vector indicating which districts to produce `studentGrowthPlots` for. If missing the function will use *all* available districts in the data to produce `studentGrowthPlots`.
- `sgPlot.schools` A vector indicating which schools to produce `studentGrowthPlots` for. If missing the function will use *all* available schools in the data to produce `studentGrowthPlots`. If both `sgPlot.districts` and `sgPlot.schools` are provided the function produces `studentGrowthPlots` for ALL students in the districts and schools provided.
- `sgPlot.reports.by.school` A boolean variable indicating whether the reports should be collated as single reports in a single folder by school and deposited into a district/school/grade folder hierarchy. The default is TRUE which puts the reports into their appropriate district/school/grade folder.
- `sgPlot.instructors` A vector indicating which instructors to produce `studentGrowthPlots` for. If NULL and the argument `sgPlot.reports.by.instructor` is TRUE, the argument function will use *all* available instructors in the data to produce `studentGrowthPlots`. If `sgPlot.districts` and/or `sgPlot.schools` are provided the function produces `studentGrowthPlots` for ALL students in the districts and/or schools provided.
- `sgPlot.reports.by.instructor` A boolean variable indicating whether the reports should be collated as single reports in a single folder by school and deposited into a district/school/grade folder hierarchy. The default is TRUE which puts the reports into their appropriate district/school/grade folder.
- `sgPlot.students` A vector of student IDs indicating which students to produce `studentGrowthPlots` for. If missing the function will use *all* available students in the data to produce `studentGrowthPlots`.
- `sgPlot.reports.by.student` A boolean variable indicating whether the reports should be collated as single reports in a single folder or deposited into a district/school/grade folder hierarchy. The default is FALSE which puts the reports into their appropriate district/school/grade slot.
- `sgPlot.header.footer.color` Character vector (default is blue) indicating the color of the header/footer associated with the `studentGrowthPlot`. Another good color is `goldenrod2`.
- `sgPlot.front.page` A path to a PDF to be used as the front page to the `studentGrowthPlot`. The default is missing so that no front page is attached to the `studentGrowthPlot`.

<code>sgPlot.folder</code>	Character vector indicating where <code>studentGrowthPlots</code> should be placed. Note that <code>studentGrowthPlots</code> are placed within nested folders within this folder. Default folder is "Visualizations/studentGrowthPlots".
<code>sgPlot.folder.names</code>	Either "name" or "number" (the default) indicating how the nested folder structure will be labeled that holds the <code>studentGrowthPlots</code> .
<code>sgPlot.fan</code>	A boolean argument (defaults to TRUE) indicating whether to produce projection fan on <code>studentGrowthPlots</code> .
<code>sgPlot.anonymize</code>	A boolean argument (defaults to FALSE) indicating whether to anonymize <code>studentGrowthPlots</code> student, school and district names. For student level anonymization, the function utilizes the <code>randomNames</code> package to produce gender and ethnicity based names based upon gender and ethnicity codes available in <code>sgp_object@data</code> .
<code>sgPlot.cleanup</code>	A boolean argument (defaults to TRUE) indicating whether to remove files produced by pdfLaTeX to produce <code>studentGrowthPlot</code> catalogs.
<code>sgPlot.demo.report</code>	A boolean argument (defaults to TRUE) indicating whether to just produce a sample <code>studentGrowthPlot</code> catalogs. Note: When producing <code>studentGrowthPlots</code> for an entire state, considerable resources are required to produce this many reports. We are actively working on parallelizing this functionality to reduce report production time by two orders of magnitude.
<code>sgPlot.produce.plots</code>	A boolean argument (defaults to TRUE) indicating whether to produce <code>studentGrowthPlots</code> . Useful when one just wants to produce wide formatted data without the actual student growth plots.
<code>sgPlot.baseline</code>	Argument (defaults to NULL) indicating whether to use baseline referenced SGPs for student growth plot construction. If not set by user, argument will be set using <code>SGPstateData</code> which contains information on whether state is a cohort or baseline referenced system.
<code>sgPlot.zip</code>	A boolean argument (defaults to TRUE) indicating whether to zip school folders containing <code>studentGrowthPlots</code> .
<code>sgPlot.output.format</code>	Argument indicating the desired type of output format for student growth plots. Either 'PDF' (default) or 'PNG'.
<code>sgPlot.show.targets.years.forward</code>	Integer or NULL argument, defaults to NULL, indicating whether growth to standard targets (currently catch-up/keep-up targets) should be calculated and represented on the student growth plot using the years forward provided.
<code>gaPlot.years</code>	A vector indicating year for which to produce <code>growthAchievementPlots</code> . If missing the function will use the last year available in the data to produce <code>growthAchievementPlots</code> .
<code>gaPlot.content_areas</code>	A vector indicating content area(s) to produce <code>growthAchievementPlots</code> using data available in <code>sgp_object</code> . If missing the function will produce plots for all available content areas provided in the data.

<code>gaPlot.students</code>	A vector of student IDs indicating which students to produce growthAchievementPlots for. If missing the function will use <i>all</i> available students in the data to produce growthAchievementPlots .
<code>gaPlot.format</code>	Either "print" or "presentation" indicating whether to optimize the plot for print form (light background) or presentation form (dark background).
<code>gaPlot.baseline</code>	Argument (defaults to NULL) indicating whether to calculate growth and achievement plots using percentile trajectories derived from baseline referenced coefficient matrices. If not set by user, argument will be set using <code>SGPstateData</code> which contains information on whether state is a cohort or baseline referenced system.
<code>gaPlot.max.order.for.progression</code>	The maximum coefficient matrix order to use for each progression. Default is NULL which utilizes the maximum order available with the coefficient matrices.
<code>gaPlot.folder</code>	Character vector indicating where growthAchievementPlots should be placed. Default folder is "Visualizations/growthAchievementPlots".
<code>parallel.config</code>	<p>A named list with, at a minimum, two elements indicating 1) the BACKEND package to be used for parallel computation and 2) the WORKERS list to specify the number of processors to be used in each major analysis. The BACKEND element can be set = to FOREACH or PARALLEL. Please consult the manuals and vignettes for information of these packages! The analyzeSGP help page contains more thorough explanation and examples of the <code>parallel.config</code> setup. TYPE is a third element of the <code>parallel.config</code> list that provides necessary information when using FOREACH or PARALLEL packages as the backend. With BACKEND="FOREACH", the TYPE element specifies the flavor of 'foreach' backend. As of version 1.0-1.0, only "doParallel" is supported. TYPE=NA (default) produces summaries sequentially. If BACKEND = "PARALLEL", the parallel package will be used. This package combines deprecated parallel packages snow and multicore. Using the "snow" implementation of parallel the function will create a cluster object based on the TYPE element specified and the number of workers requested (see WORKERS list description below). The TYPE element indicates the users preferred cluster type (either "SOCK" for socket cluster or "MPI" for an OpenMPI cluster). If Windows is the operating system, this "snow" implementation must be use and the TYPE element must = "SOCK". Defaults are assigned based on operating system if TYPE is missing based on system OS. Unix/Mac OS defaults to the "multicore" to avoid worker node prescheduling and appears to be more efficient in these operating systems. The WORKERS element is a list with GA_PLOTS (growth achievement plots) and SG_PLOTS (student growth plots) specifying the number of processors to be used. NOTE: choice of the number of cores is a balance between the number of processors available and the amount of RAM a system has; each system will be different and may require some adjustment.</p> <p>Default is FOREACH as the back end, TYPE=NA and both plot WORKERS=1, which produces plots sequentially: <code>'list(BACKEND="FOREACH", TYPE=NA, WORKERS=list(GA_PLOTS=1, SG_PLOTS=1))'</code></p>

Examples of various parallel configurations can be found in the examples for [analyzeSGP](#) and [summarizeSGP](#).

Value

Function produces *numerous* (potentially hundreds of thousands) of pdf files in a folder structure specified by the user and supplied through arguments to the function.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <vaniwaarden@colorado.edu>

References

Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.

Betebenner, D. W. (2009). Norm- and criterion-referenced student growth. *Educational Measurement: Issues and Practice*, 28(4):42-51.

See Also

[bubblePlot](#), [bubblePlot_Styles](#), [studentGrowthPlot](#), [growthAchievementPlot](#)

Examples

```
## Not run:
## visualizeSGP is Step 5 of 5 of abcSGP
Demonstration_SGP <- sgpData_LONG
Demonstration_SGP <- prepareSGP(Demonstration_SGP)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP)
Demonstration_SGP <- combineSGP(Demonstration_SGP)
Demonstration_SGP <- summarizeSGP(Demonstration_SGP)
visualizeSGP(Demonstration_SGP)

## Production of sample student growth and achievement plots

visualizeSGP(
  sgp_object=Demonstration_SGP,
  plot.types="studentGrowthPlot",
  state="DEMO",
  sgPlot.districts=470,
  sgPlot.schools=c(6418, 8008),
  sgPlot.header.footer.color="#4CB9CC")

## End(Not run)
```

Index

*Topic **classes**

SGP-class, [34](#)
splineMatrix-class, [36](#)

*Topic **datasets**

SGPstateData, [36](#)

*Topic **documentation**

abcSGP, [4](#)
analyzeSGP, [8](#)
baselineSGP, [15](#)
capwords, [24](#)
combineSGP, [25](#)
createKnotsBoundaries, [27](#)
gofSGP, [28](#)
outputSGP, [31](#)
prepareSGP, [33](#)
summarizeSGP, [56](#)
testSGP, [60](#)
updateSGP, [61](#)
visualizeSGP, [63](#)

*Topic **misc**

bubblePlot, [17](#)
bubblePlot_Styles, [22](#)
growthAchievementPlot, [29](#)
studentGrowthPercentiles, [38](#)
studentGrowthPlot_Styles, [49](#)
studentGrowthProjections, [51](#)

*Topic **models**

bubblePlot, [17](#)
bubblePlot_Styles, [22](#)
growthAchievementPlot, [29](#)
studentGrowthPercentiles, [38](#)
studentGrowthPlot_Styles, [49](#)
studentGrowthProjections, [51](#)

*Topic **package**

SGP-package, [3](#)

abcSGP, [4](#), [30](#), [33](#), [61](#), [62](#)
analyzeSGP, [3–7](#), [8](#), [15–17](#), [25](#), [26](#), [30](#), [33–35](#),
[54](#), [56](#), [58](#), [59](#), [68](#), [69](#)
as.splineMatrix (splineMatrix-class), [36](#)

baselineSGP, [15](#)

bubblePlot, [3](#), [7](#), [17](#), [22–24](#), [64](#), [65](#), [69](#)

bubblePlot_Styles, [22](#), [69](#)

capwords, [24](#)

combineSGP, [3–7](#), [12](#), [17](#), [25](#), [33](#), [34](#), [56](#), [59](#)

createKnotsBoundaries, [27](#)

gofSGP, [28](#)

growthAchievementPlot, [3](#), [7](#), [29](#), [64](#), [67–69](#)

is.SGP, [34](#)

is.SGP (SGP-class), [34](#)

is.splineMatrix (splineMatrix-class), [36](#)

outputSGP, [3](#), [4](#), [31](#)

prepareSGP, [3–9](#), [12](#), [15](#), [17](#), [25](#), [26](#), [30](#), [33](#),
[33](#), [35](#), [56](#), [57](#), [59](#), [62](#)

SGP (SGP-package), [3](#)

SGP-class, [34](#)

SGP-package, [3](#)

sgpData, [4](#), [39](#), [43](#), [52](#), [55](#)

sgpData_INSTRUCTOR_NUMBER, [7](#)

sgpData_LONG, [5](#), [33](#), [34](#), [43](#)

SGPstateData, [25](#), [36](#), [41](#), [43](#)

splineMatrix-class, [36](#)

studentGrowthPercentiles, [3](#), [4](#), [7](#), [10](#), [12](#),
[16](#), [28](#), [29](#), [36](#), [37](#), [38](#), [39](#), [43](#), [52](#), [55](#)

studentGrowthPlot, [3](#), [7](#), [47](#), [49](#), [51](#), [64](#), [66](#),
[67](#), [69](#)

studentGrowthPlot_Styles, [49](#)

studentGrowthProjections, [3](#), [4](#), [7](#), [10](#), [12](#),
[38](#), [43](#), [51](#)

summarizeSGP, [3–7](#), [17](#), [22](#), [33–35](#), [56](#), [62](#), [64](#),
[69](#)

testSGP, [60](#)

updateSGP, [61](#)

visualizeSGP, [3–7](#), [30](#), [34](#), [49](#), [63](#)